DEFECT AND COMPONENTS RECOGNITION IN PRINTED CIRCUIT BOARDS USING CONVOLUTION NEURAL NETWORK

CHEONG LEONG KEAN

UNIVERSITI SAINS MALAYSIA

2018

DEFECT AND COMPONENTS RECOGNITION IN PRINTED CIRCUIT BOARDS USING CONVOLUTION NEURAL NETWORK

by

CHEONG LEONG KEAN

Thesis submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Engineering (Mechatronics Engineering)

JUNE 2018

ACKNOWLEDGEMENT

First and foremost, I would like to thank Assoc. Prof. Dr. Shahrel Azmin bin Suandi for the constant guidance for this final year project (FYP). He inspired me with the beauty of knowledge in machine learning, allowing me to make the decision to take up Convolutional Neural Network based project. Thankfully, the project is successfully completed. The discovery and exploration in this project has given me insights on how to apply machine learning algorithms in the industry.

Besides that, special thanks are directed to Vitrox, an automated inspection vision solution company in Penang island for providing the title for this project and printed circuit boards images. Gratitude are also directed to Ooi Song Wah, Kang Faydi and Saimunur Rahman which are my mentors in this project.

Special appreciation to the School of Electrical and Electronic Engineering (PPKEE) for providing usage of the computer in communication lab. The hardware support from the school plays an important role in experimentations of this project and therefore it is really appreciated.

In addition to that, I am very grateful to Google's researcher in providing open source Tensorflow API which are mainly used for deep learning research. Detailed and extensive tutorials are also provided to guide me from finishing this project.

Finally, I would like to like to give a final thanks to my family and friends who has always supported me. Advices and encouragement from them are always motivation for me to go through the hard times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CNN	Convolutional neural network
COCO	Common Objects in Context
CPU	Central Processing Unit
FN	False Negative
FP	False Positive
GPU	Graphical Processing unit
HOG	Histogram of Oriented Gradient
ILSVRC	Imagenet Large Scale Visual Recognition Challenge
K-NN	K-Nearest Neighbour
mAP	Mean Average Precision
ML	Machine Learning
MLP	Multi-Layer Perceptron Neural Network
PCANet	Principal Component Analysis Network
PCB	Printed Circuit Board
PCB-CC	PCB Component Classifier
R-CNN	Region based Convolution Neural Network
ROI	Region of Interest
SIFT	Scale-Invariant Feature Transform
SSD	Single Shot Multi-box Detector
SVM	Support Vector Machine
TEM	Texture Energy Measures
TN	True Negative
TP	True Positive
YOLO	You Only Look Once

SISTEM PENGECAMAN KOMPONEN DAN KECACATAN PAPAN LITAR BERCETAK MENGGUNAKAN RANGKAIAN NEURAL KONVOLUSI

ABSTRAK

Pertumbuhan peranti elektronik meningkatkan permintaan pengeluaran papan litar bercetak, "printed circuit coards" (PCB) dalam industri elektronik. Ini menjurus kepada kenaikan kuantiti pengeluaran PCB setiap hari. Oleh itu, pemeriksaan visual automatik menjadi sistem penting untuk dilengkapkan di mana-mana barisan pengeluaran supaya kualiti PCB dihasilkan dapat dipastikan. Matlamat projek ini, ialah membina sistem pengecaman kecacatan dan komponen automatik untuk PCB menggunakan rangkaian neural konvolusi, "convolution neural network" (CNN). Skop projek ini adalah untuk melaksanakan pengesanan dan kecacatan pengesanan komponen PCB. Pada peringkat pertama, model CNN yang sesuai akan dibina untuk membezakan komponen-komponen elektrik di atas papan litar bercetak. Untuk menjimatkan masa, pemindahan pembelajaran dengan model yang sudah terlatih seperti VGG16, DenseNet169 dan InceptionV3 telah dilakukan untuk mengkaji model yang sesuai untuk pengiktirafan komponen. Menggunakan pembelajaran pemindahan dengan VGG-16, hasil terbaik dicapai adalah ketepatan 99% dengan kemampuan untuk mengiktiraf 25 komponen yang berbeza. Selepas itu, penyetempatan objek dilakukan dengan menggunakan rangkaian neural convolutional berasaskan rantau "region based convolution neural network" (R-CNN). Pelbagai eksperimen telah dilakukan untuk menentukan kaedah dan parameter latihan yang optimum untuk mencapai sistem yang dapat mengesahkan kecacatan pada PCB dengan ketepatan yang tinggi. "Mean Average Precision" (mAP) terbaik yang dicapai untuk sistem pennempatan kecacatan ialah 96.54%.

DEFECT AND COMPONENTS RECOGNITION IN PRINTED CIRCUIT BOARDS USING CONVOLUTION NEURAL NETWORK

ABSTRACT

The growth of electronic devices increases the demands of printed circuit boards (PCB) productions in the electronic industries. This leads to the rise in the quantity of PCB productions every day. Consequently, automated visual inspection becomes an essential system to be equipped in any production line to ensure the quality of the PCB produced which brings us to the aim of this project, building an automated components recognition system for PCB using CNN. In addition to that, localization on the defects of the PCB components will also be performed. In the first stage, a simple CNN-based component recognition classifier will be developed. Since training a CNN from scratch is expensive, transfer learning with ImageNet pre-trained models is performed instead. Pre-trained models such as VGG16, DenseNet169 and InceptionV3 are used to investigate which model suits the best for components recognition. Using transfer learning with VGG-16, the best result achieved is 99% accuracy with the capability of recognizing up to 25 different components. Following that, object localization is performed using faster region-based convolutional neural network (R-CNN). Multiple experiments have been performed to determine the optimum method and training parameters to achieve a system that is able to localize defects on the PCB with high accuracy and precision. The best mean average precision (mAP) achieved for the defects localization system is 96.54%.

CHAPTER 1

INTRODUCTION

1.1 Background

The printed circuit board (PCB) manufacturing is getting more and more important because nowadays, a lot of consumer electronics products, such as laptops, mobile phones, tablets, PCs and so on are the essentials in our daily life (Malge, 2014) . With the huge demands of PCBs, manufacturers need to cope with the demands and produce large quantity of PCBs. This situation arises a question, which is how do these manufacturers ensure the quality of their PCBs, but still produce large quantities of them? The solution to that is through an automated quality inspection system. It is an approach to counter difficulties occurred in human's manual inspection which will have a lot of error when the quantity of PCB to be inspected increases in bulk. Automated visual PCB inspection can provide fast and quantitative information of defects which makes it very popular and important in the manufacturing process. A few examples of defect detection methods on PCB can be seen in Malge (2014) and Takada et al. (2017) papers.

There are three main processes in PCB inspection which are defect detection, defect classification and defect localization. Currently there are many algorithms developed for PCB defect detection and classification using contact or non-contact methods (Moganti et al., 1996). However, non-contacts methods are popular in literature because they come in a wide range of selection from x-ray imaging, ultrasonic imaging, thermal imaging to optical inspection using image processing. Although these methods are successful in detecting defects, but they are not very effective in classifying the defects. In recent years, machine learning (ML) driven solutions has shown promising results in visual classification and localization. There are many different ML approaches to differentiate between images however, convolutional neural network (CNN), a deep artificial neural network (ANN) technique stands out the most. It has excellent recognition capabilities to recognize handwriting (Suryani et al., 2017), medical image processing (Harangi et al., 2017) and video classification (Burney and Syed, 2016). Moreover, In ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) where for many years winning team uses CNN as its core.

While majority of the works in CNN are for image classification, there are some researchers uses it for advance applications such as image localization. For example,

recent advances in object detection are driven by the success of region proposal methods (Uijlings et al., 2012) and region-based convolution neural networks (Girshick et al., 2014). Although region-based convolutional neural networks (R-CNN) were computationally expensive as originally developed in Girshick et al. (2014) their cost has been drastically reduced; thanks to the sharing of convolutions across proposals (Ren et al., 2017). In ImageNet Large Scale Visual Recognition Competition (ILSVRC) and Common Objects in Context (COCO) 2015 competitions, Faster R-CNN and region proposal network (RPN) were the core of several first place entries (He et al., 2016) in the tracks of ImageNet detection, ImageNet Localization, COCO detection, and COCO segmentation. RPNs can learn to propose regions from data, and therefore easily benefit from deeper and more expressive features (for example, the 101-layer residual nets adopted in He et al. (2016).) These results suggest that faster R-CNN is an effective and accurate object detector.

In this project, the aim is to develop an automated visual inspection system for PCB components. At the beginning, the camera will take images of the PCB. Then these images are fed into the defect identification system the PC to analyse. A CNN based approach is developed with the ability to recognize and localize the components on the PCB and determine whether there is any defect or not. The typical challenges of this projects are to cater for the illumination, colour, rotation variations in the PCB component images.

1.2 Problem Statement

The development of consumer electronic products increases everyday as new products are being developed every year. As a result, the production of printed circuit board free of defects is highly demanded. According to Anitha and Mahesh (2017) a huge number of different techniques and algorithms was proposed in literature for PCB defect analysis Crispin and Rankov (2007) used template-matching approach to detect the PCB components. Raihan and Ce (2017) used OPENCV with image subtraction method for PCB defect detection. Besides that, Ganavi and Rao (2016) used image processing techniques such as template matching, wavelet transform and background subtraction to detect defects such as missing or wrongly placed components, broken or shorting of the PCB tracks. From these existing detection frameworks, it can be inferred that each algorithm is limited to detect only a specific type of defects. With that being said, recently,

convolutional neural network (CNN) has achieved excellent performance on machine vision tasks, particularly image recognition problems (Hosseini et al., 2017). Hence, convolutional neural network is proposed to be used in this project for components recognition and defects localization.

1.3 Objectives

The main goal of this project is to develop a deep learning-based components and defects recognition system for PCB with the following objective

1. To develop a CNN classifier to recognize different types of electrical components on the PCB.
2. To compare and identify the best CNN models among Inception V3, VGG-16 and Densenet-169 in classifying the electrical components on the PCB board using transfer learning.
3. To perform localization of the defects on the PCB using faster R-CNN.

1.4 Project Contribution

The aim of this project is to propose a CNN detection framework to replace most of the existing frameworks that uses handcrafted features. The CNN classifier can be used to automatically inspect and find the defects in PCB.

1.5 Project Scope

The project mainly aims to detect the defects and components on the printed circuit board (PCB) using convolutional neural network (CNN) and then classify them. Several limitations are assumed in this project. They are:

1. Training of CNN has been performed under limited datasets because most of the industrial company will keep their PCB designs and images confidential, thus not willing to share much of their images.
2. Detection and classification have been done under controlled light condition with minimal interference of ambient light.
3. Classification of components and defects are only applicable at the same scale which is the size of components for classification are of one size only.
4. Training of the CNN model is done using transfer learning method which is a method of reusing pre-trained models to train the new custom datasets.

1.6 Thesis Outline

A total of five chapters are included in this thesis, each containing detailed explanation of the project. Chapter one starts with the background research, followed by the problem statement, objectives, project contributions and then the project scope.

Chapter two which is the literature review, contains a thorough review of previous work on image classification algorithms. Different kinds of existing solutions are discussed to investigate the problems and limitations. Moreover, a summary for the literature review is also performed to get the proposed solution for methodology.

The third chapter, methodology explains the method of training the convolutional neural network (CNN) in details. The transfer learning method on DenseNet-169, VGG-16 and Inception V3 is further explained. Besides that, the dataset evaluation method which is to evaluate performance of the model is also explained.

The fourth chapter is the results and discussions. In this chapter, the performance of the model is evaluated and tabulated as results. The credibility of the model itself can be seen based on the accuracy obtained from testing.

Chapter five is the conclusion. The whole thesis is summarised, and future development and suggestion of the project is mentioned.

CHAPTER 2

LITERATURE REVIEW

This chapter will focus on the different classification methods of objects, medical based images and industrial related defects. In section 2.1, classification methods using handcrafted features are discussed where several algorithms to extract features are introduced. Besides that, the method of classification using hybrid of handcrafted features and non-handcrafted features (deep learning features) will be explained too. Section 2.2 will then discuss about the different deep learning methods in classification. The method of localizing objects which are faster R-CNN and single shot multi-box detectors (SSD) are discussed in section 2.3. Following on, section 2.4 discusses about transfer learning and section 2.5 discusses about data augmentation, a method to expand datasets.

2.1 Handcrafted Features

Handcrafted features are properties of images, usually found using algorithms such as scale-invariant feature transform (SIFT) and Laws Texture Energy Measures (TEM). The features obtained are usually used with common machine learning approaches like support vector machine (SVM) or K-Nearest Neighbour (K-NN) for object recognition or machine vision. A research has been done by (Acar and Özerdem, 2013) using Laws TEM to extract the features and then separate them using K-NN classifier to recognize humans based on their eye images which has different iris texture. It is seen that the performance of his project reaches up to 80.74% for 2 classes which is the classifier can identify up to 2 individuals. The accuracy of the classifier decreases to only 60.74% in 10 neighbours structure of K-NN (Acar and Özerdem, 2013). Another approach for classification using handcrafted features is made by Kirti and Bharat using SIFT as the features for the SVM training (Kirti and Bharat, 2016). Hybrid SIFT-SVM obtained an accuracy up to 81.2% in recognizing child, adult and old faces. One of the limitation of using hand-crafted features is that one would have to extract the features by himself. Choosing the correct features is no easy task as matching the wrong features together for training would get an undesirable result.

2.1.1 Hybrid of Handcrafted Features and Non-Handcrafted Features

Hybrid of handcrafted features and non-handcrafted features is actually explored too in order to achieve better accuracy. Using this method, handcrafted features are mixed with automatically extracted features through deep learning and used with the typical classifiers like SVM or K-NN. A new robust vehicle classification method is proposed where it uses the features extracted through a deep learning network which is principal component analysis network (PCANet), handcrafted features of Histogram of Oriented Gradient (HOG) and Hu moments are extracted then combined and input into SVM for classification (Jiang et al., 2017). Results shows that the Hybrid method is able to achieve up to 98.34% accuracy, exceeding the PCANet model classifier by 5.14% which has an accuracy of 93.20% (Jiang et al., 2017). From this research, it is clearly seen that this hybrid method can attain comparable or higher performance compared to deep learning methods. However, certain limitations are present using this method, where in the paper (Jiang et al., 2017), the dimensions of the features are relatively high, causing the process of classification to be time consuming. Besides that, in another research who also employed the method of combining deep learning and hand-crafted features for skin lesion classification, states that the limitation of the project was the various image distortion and artefacts that are present in input images causes the accuracy to be slightly lower (Majtner et al., 2017).

2.2 Deep Learning

Deep learning is a method which uses complex models that consists of multiple hidden layers that can extract features from data and learn them. The deep learning methods have significantly improved speech recognition, object recognition and detection, and many other domains (Lecun et al., 2015).

2.2.1 Convolutional Neural Network (CNN)

Convolutional neural network is one of the deep learning models that performs well in processing images and videos (Arel et al., 2010). As shown in Figure 2.1, a CNN consists on an input and an output layer, with multiple hidden layers. The hidden layers are either convolutional, pooling or fully connected. In the convolution layers, we apply a convolution operation to the input, passing the result to the next layer. For example, we apply a weight matrix which behaves like a filter to extract information such as edges, corners and so on. A pooling layer is applied after convolution layer. There are several

layer options which one of them is max-pooling. This basically takes a filter and strides along the input images and outputs the maximum number in every sub-region that the filter convolves around. Finally, fully connected layer is the same as the traditional multi-layer perceptron neural network (MLP) which consists of network of neurons to calculate the classification score.

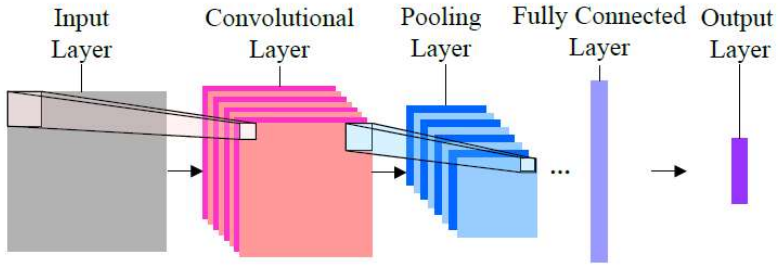


Figure 2.1: Simple structure of a convolutional neural network (CNN) (Peng et al., 2016).

2.2.2 Convolutional Neural Network for Machine Vision

Currently there are a lot of researches and experiments that involves convolutional neural network. In 2016, a CNN architecture is proposed for automated feature extraction in industrial inspection by Weimer et al. (2016). Besides that, an automatic defect recognition in X-ray testing using computer vision is proposed in the year of 2017 Mery and Arteta (2017). Masci et al. (2012) proposed a Max-pooling CNN approach for supervised steel defect classification. Another CNN approach for crack damage detection is proposed by Cha et al. (2017). Soukup and Huber-Mork (2014) proposed using CNN for steel surface defect detection for example rail defects. Another CNN approach to detect cracks on nuclear power plant components is proposed by Chen and Jahanshahi (2017). Using CNN approach, crack with noisy patterns and tiny crack with low contrast and variant brightness can be detected Chen and Jahanshahi (2017). An automated tunnel inspection solution using deep convolutional neural network is also proposed by Makantasis et al. (2015). From all of these proposed solutions it is clear that there are already a lot of defect detection systems that are successful using CNN. Hence, CNN would be a great approach in detecting detailed defects in the components of a PCB board and is proposed in this project.

2.3 Localization with Deep Learning

Besides these CNN methods for machine vision inspections, there are actually other CNN approaches that can perform localization of objects which is very helpful in detecting components defects on PCB. They are faster region-based convolutional neural network (R-CNN) and single shot multi-box detectors (SSD).

2.3.1 Faster R-CNN

The object detection system called, faster R-CNN consists of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector that uses the proposed region (Girshick, 2015). The entire system then becomes a single unified network for object detection as shown in Figure 2.2. The region proposal network (RPN) tells the Fast R-CNN module where to look which proceeds to detect objects in an image (Ren et al., 2017). Fang et al. (2017) used faster R-CNN to perform detection on microscopic cells mentioned the limitation of the project is that more data is needed to achieve better results.

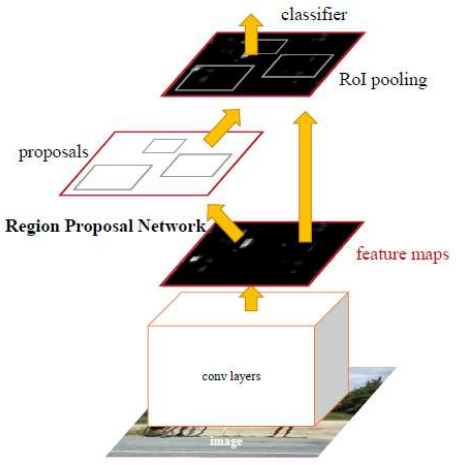


Figure 2.2: Faster R-CNN as a single, unified network (Ren et al., 2017).

2.3.2 Single Shot Multi-Box Detector (SSD)

The SSD is based on a feed-forward CNN that can forms a collection of bounding boxes and scores for the presence of object in those boxes and then finally classify the object. The SSD detector normally starts with a pre-trained model that is converted to a fully convolutional neural network. In Liu et al. (2016) research, VGG-16 network is used

as the base but the fully connected layers are changed with additional convolutional layers. One of them is a multi-scale feature maps for detections are added which would decrease in size progressively to detect objects in multiple scales. Besides that, a convolutional predictor to predict the object is also added to prediction for every multi-scale feature map.

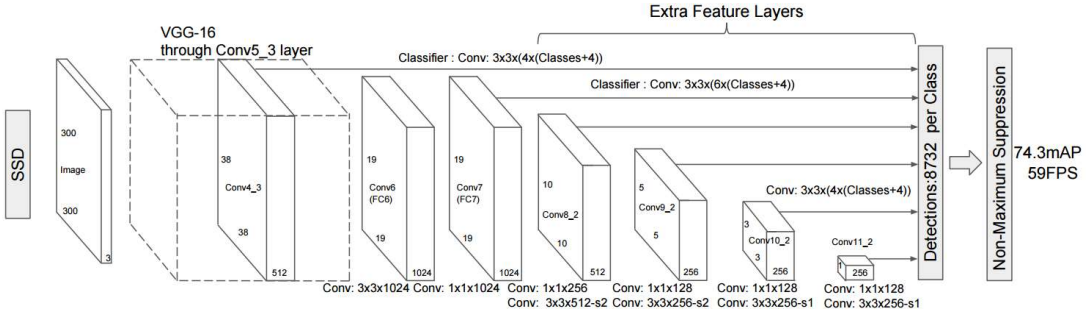


Figure 2.3: Single shot multibox detector (SSD) architecture (Liu et al., 2016).

SSD matches objects with default boxes (anchors) of different aspects as shown in the dashed rectangles in Figure 2.4. Each element of feature map has several default boxes associated with it. If any default box with an Intersection Over Union/Jaccard index (IoU) of 0.5 or greater with a ground truth box is considered as an object. In Figure 2.8 two of the 8x8 boxes are matched with the cat (blue box), and one of the 4x4 boxes is matched with the dog (red box) (Liu et al., 2016).

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{2.1}$$

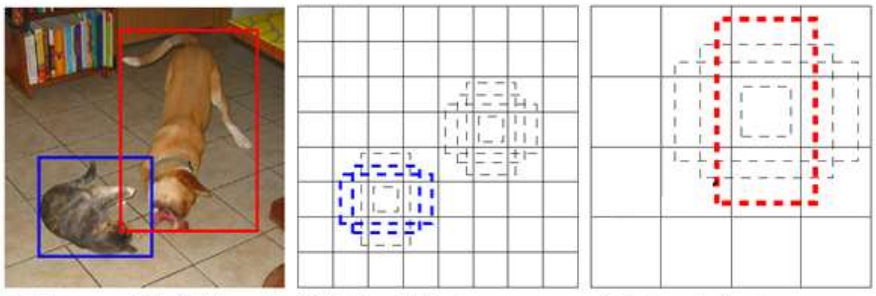


Figure 2.4: SSD default box matching (Liu, et al., 2016).

There are a few examples of research using SSD. A research using SSD with transfer learning for ship detection using Chinese Gaofen-3 images was made by Wang et al. (2017). Besides that, Ning et al. (2017) used an inception styled SSD for object detection. SSD might be one of the fastest algorithm but comes with limitation; there is a big gap between the current accuracy of other algorithms such as faster R-CNN (Ning et al., 2017). For faster inference speed, one would choose SSD with some compensation of accuracy. For higher accuracy, one would choose faster R-CNN with some compensation of the speed of inference.

2.4 Pretrained-Models

A pre-trained model is a model which is already created by someone and trained with existing datasets. For example, there are pre-trained models like VGG-16, DenseNet-169 and Inception V3 which are trained with ImageNet dataset or Common Objects in Context (COCO) dataset. Instead of spending years to build a decent image recognition algorithm from scratch, a pre-trained model can be used to recognize images using the transfer learning method which is mentioned in Section 2.5. A pre-trained model can give a high accuracy of recognition while saving the huge effort required to build a model from scratch.

2.4.1 VGG-16

The model of VGG-16 shown in Figure 2.5, consists of 16 layers consisting of the input layers, convolutional layers and fully connected layers which are separated in groups by the max pooling layers. The many convolution layers of the VGG-16 allows the model to extract very “deep” features from the input images. The features extracted from the “deeper” layers greatly increases the accuracy of the model which is why VGG-16 is useful (Chen-McCaig et al., 2017). In Simonyan and Zisserman (2015) paper it is seen that the very deep Convolutional networks (VGG networks), clearly outperforms a lot of previous generation models.

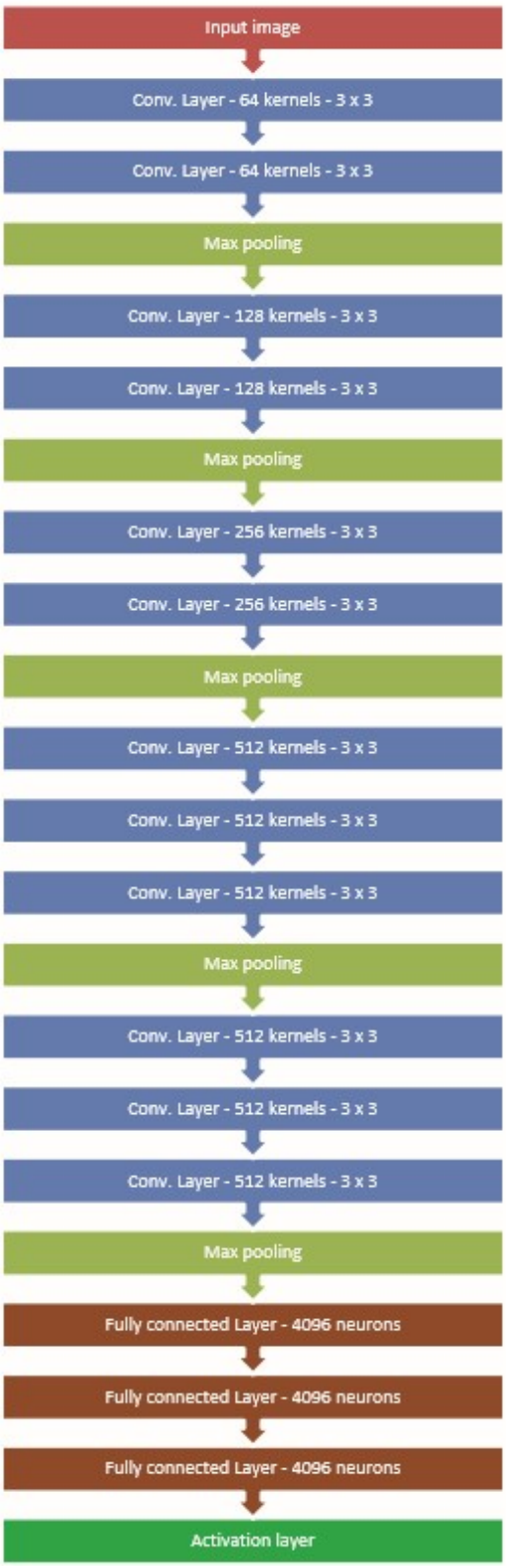


Figure 2.5: VGG-16 network structure (Chen-McCaig et al., 2017).

2.4.2 DenseNet

A Dense Convolutional Network (DenseNet), is a network that connects each layer to every other layer in a feed-forward fashion as shown in Figure 2.6. For each layer, the feature maps obtained from all the layer behind it are used as input. Moreover, its own feature maps are used as inputs for the layer in front. This method of connection allows the information flow between layers to further improve. The network configurations of different DenseNet can be seen in Figure 2.6. It is said that DenseNet have bring forth several advantages which are they are able to reduce severity of the vanishing-gradient problem, strengthen feature propagation, allow feature reusing, and finally greatly reduces the number of parameters (Huang et al., 2017). In Kreso et al. (2017) paper, DenseNet-169 model is used to perform semantic segmentation of large natural images and it can be seen that the DenseNet-169 used performs well reaching a pixel accuracy of 95.11%. Besides that, there is another research to detect and classify Alzheimer's disease using DenseNet shows outstanding results of classifying by having a high 99% precision DenseNet-169 model (Islam and Zhang, 2017).

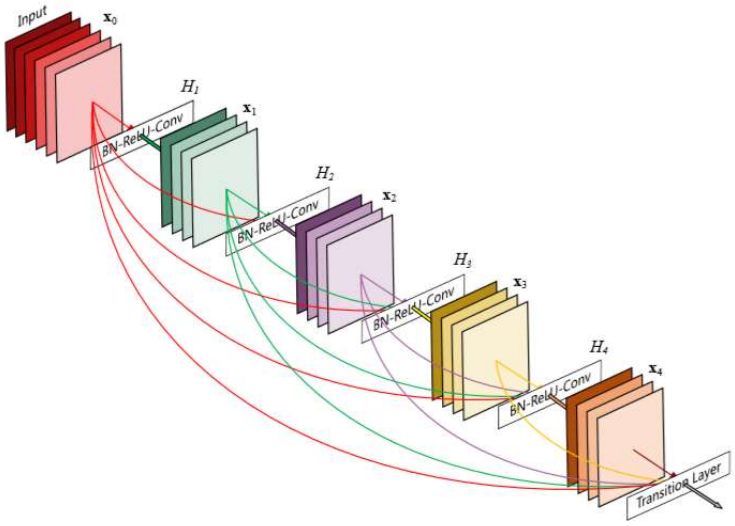


Figure 2.6: A 5-layer dense block with each layer taking all preceding feature-maps as input (Huang et al., 2016).

Table 2.1: DenseNet architecture for ImageNet (Huang et al., 2016).

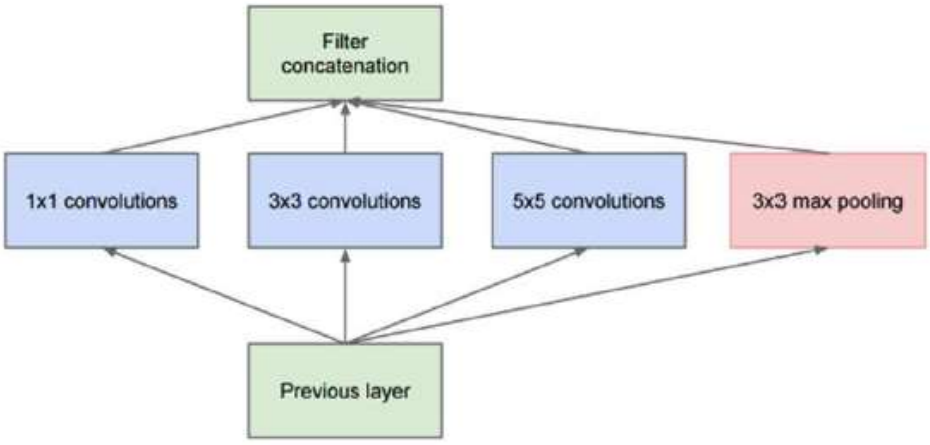
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

2.4.3 Inception

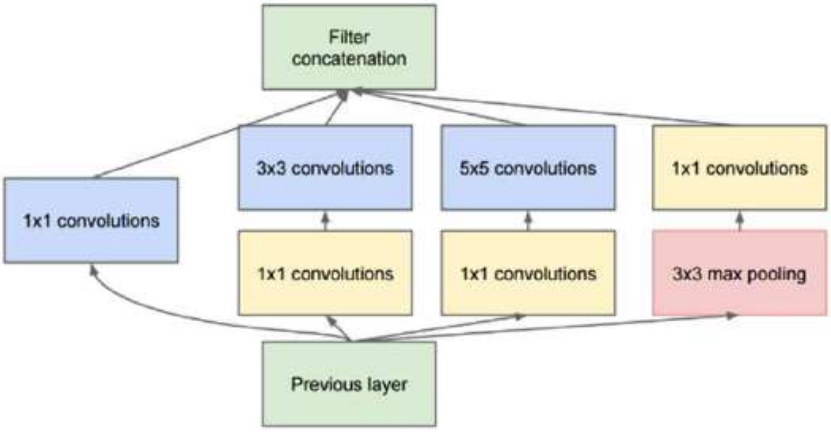
Christian Szegdy from Google wanted to develop a method to reduce the computational demands of deep neural network. This has led to the creation of inception architecture also known as GoogleNet. The inception is so good that it can perform even under poor computational power (Szegedy et al., 2015). It is also known that Inception has a much lower computational cost than VGGNet (He et al., 2015), which allows a huge amount of data to be used in the Inception model without worrying of needing a high computational capacity (Schroff et al., 2015).

The basic architecture of the Inception module can be seen in Figure 2.7 (a). The Inception module consists of parallel convolutions that comes with different kernel sizes, which at the output, they are concatenated. This means that all the output from the different convolutional layers are stacked together. Generally, the Inception module uses 1x1, 3x3 and 5x5 convolution layer with a 3x3 max pooling layer prior to the concatenation filter.

However, the large convolution layer used, particularly the 5x5 convolution layer, produces a large amount of feature maps, resulting in a computationally expensive model. This type of model can be seen in the naïve version of the Inception module. The solution to this problem would be by using the 1x1 convolution layer as shown in Figure 2.7 (b). The dimensionality of the input before entering the large convolution layer will then be reduced and subsequently reducing the computational requirement too.



(a) Inception module, naïve version



(b) Inception module with dimension reductions

Figure 2.7: Different Inception module (Szegedy et al., 2015b).

2.4.4 Inception V3

Inception V3 which is an improved version of the initial structure of Inception V1 and Inception V2, is one of the many pre-trained models available at the internet as a free resource to be used. From Figure 2.8, it can be seen that the inception layers of the inception V3 comprised of convolutional layers and pooling layers which are cascaded and then concatenated at the end (Chen-McCaig et al., 2017). It is also reported that Inception V3 is an outstanding CNN model which works really good in classifying image (Szegedy et al., 2015b). Several researches also used Inception V3 as pretrained models to train their own custom data and showed very high accuracy. For example, Xia et al. (2017) performed classification of 28 different flower species and is able to obtain a high accuracy of 95% with their method. Besides that, another research, also used Inception V3 as its pretrained model to classify medical images to determine breast cancer. The inception V3 consists of many hidden layers which is also known as the inception layers (Chang et al., 2017).

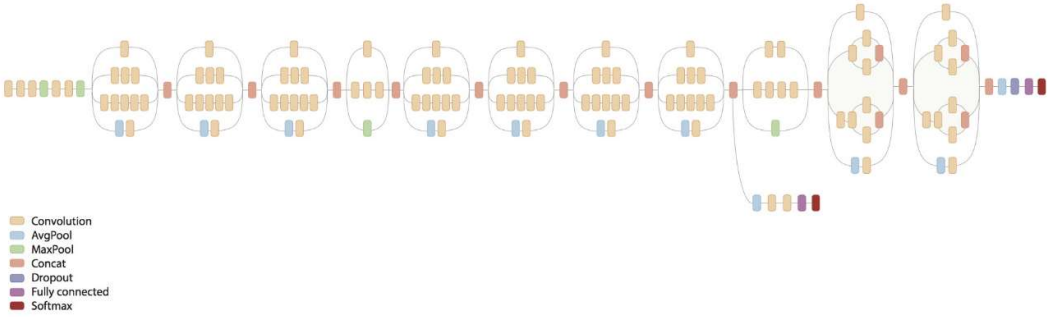


Figure 2.8: Inception V3 model (Chen-McCaig et al., 2017).

2.5 Transfer Learning

Many researchers have found that neural nets (NN) pre-trained on one task can learn new related tasks more quickly and better (Ciresan et al., 2012). Therefore, transfer learning is applied on pre-trained models to gain accurate results and yields faster training time. Figure 2.10 is the illustration of the how transfer learning is performed. Model A will be trained with a large dataset of object images. Then the knowledge learned will be transferred to a new model which is for example, model B. Due to the knowledge learnt previously from model A, is transferred to model B, model B can learn new but similar

types of images with just a relatively small custom dataset and still achieving high accuracy.

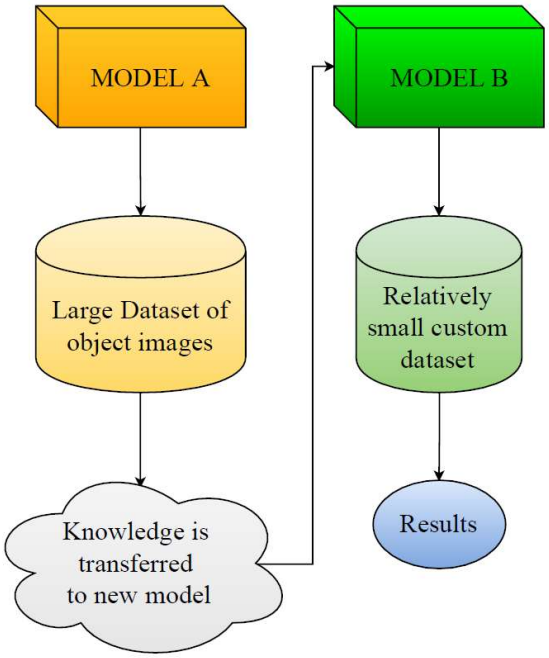


Figure 2.9: Transfer learning concept.

2.6 Data Augmentation

Often, datasets are limited. The insufficient number of datasets is always a problem for deep neural network which can only perform well when large amount of data is used for training. Therefore, data augmentation method is often used to increase the datasets and even creating "artificial conditions" to increase the accuracy. Artificial conditions can refer to different angle, different focus of image, different orientation of the object, different lighting conditions, blurring, noise and so on. These different conditions will allow the neural network to learn to recognize images in many ways and thus, increasing the accuracy. In this paper Dellana and Roy (2016), data augmentation is used to solve the overfitting problem of CNN. It is seen that data augmentation can indeed help in improving the accuracy provided that the correct augmentation method is chosen for a particular type of dataset. However, if the wrong method is chosen, accuracy might decrease. Figure 2.11 shows the different ways of augmentation of data which can effectively increase the number of datasets.

Figure 2.10: Different types of data augmentation applied to an image (Dellana and Roy, 2016).

2.7 Summary

Based on the literature readings, there are a variety of algorithms to classify images. For example, classification in support vector machine (SVM) using handcrafted features, hybrid of handcrafted features and non-handcrafted features. Currently, there are very little research on papers which uses CNN to perform classification on components and defect detection on PCB. However, there are still many different CNN algorithms used to classify medical images, defects and objects with high accuracy. Hence, it is evident that this project should use CNN to classify the components and defects as well. Transfer learning method will also be employed so that an accurate classifier can be produced even with limited datasets. Three different pre-trained models, Inception V3, DenseNet-169 and VGG-16 will be investigated to find the suitable model to classify the components. Besides that, data augmentation method is also used to increase the number of datasets to allow an effective training on the CNN. Faster R-CNN and SSD will be experimented to localize the defects on the PCB board.

CHAPTER 3

METHODOLOGY

3.1 Overview

The overview of this chapter includes the whole process of developing a component recognition classifier and localization of defects on PCB board. Currently, there are no publicly available PCB components dataset. Therefore, the datasets which consist of PCB images are collected from Vitrox, which is a company focusing on manufacturing automated vision inspection system.

This project is divided into two parts. The first is to develop a CNN model which can classify different types of components. In Figure 3.1, the overall flow of the training process for the CNN model classifier can be seen. One important step before starting the training of the CNN model is the data preparation step. In the data preparation step, the images are first cropped to get the region of interest (ROI) which is the components on the PCB. Then, these images are segregated into three different parts which are training, validation and testing data. Data augmentation is also performed to increase the number of training images. After the dataset preparation step, the images are then fed into the CNN for training. However, with the limited number of datasets obtained, training will not be effective, and the classification accuracy will be low. Therefore, a method called transfer learning is applied in the training process. Using this method, only the classification layer of the pre-trained model is retrained instead of the full network. In this project, the model, VGG-16, DenseNet-169 and InceptionV3 is used to compare the performance between them. Next, based on the accuracy of the training and validation graph plotted through the training process, it can be determined if the model is matured or not. If the accuracy of the training and validation graph converges to a very high percentage as training epoch increases, then we can consider the model is mature and move on to the next step. Model evaluation will then be performed using train-test split and K-cross validation. If consistent high accuracies are achieved in each evaluation, then we archive the model. The archived model is the final model trained to classify the components with the best accuracy.

The second part of the project is localization of the defects. One of the methods proposed is faster Region-based convolution neural network (R-CNN). The basic process of training a faster R-CNN network is shown in Figure 3.2. It is almost similar to the

normal CNN training except an additional step. After the images are prepared, the defects on the images are labelled first. Then the images along with their respective labels are used for training. By having the label of the images in the training, the model will be able to find the region of interest and try to learn it.

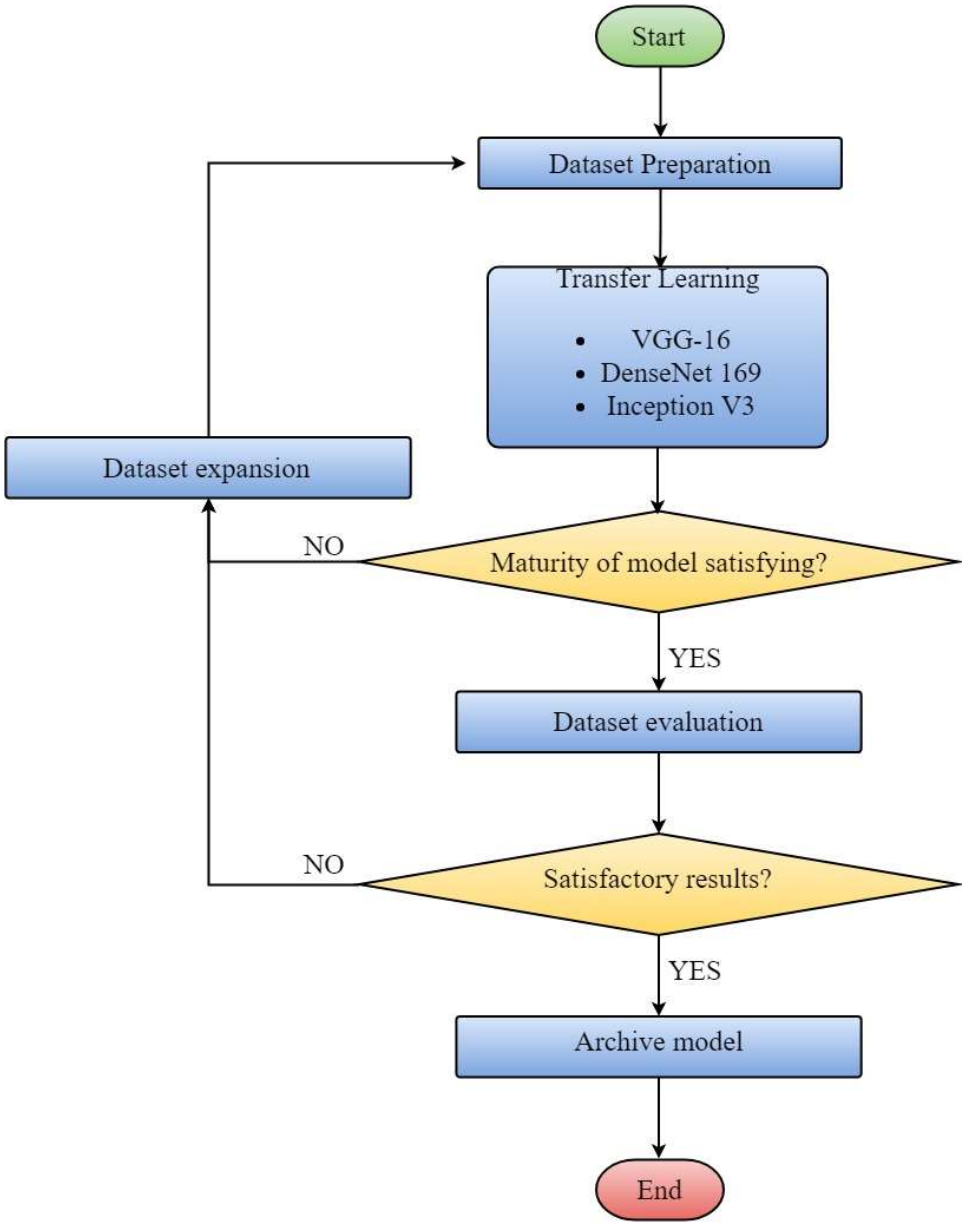


Figure 3.1: Overall flowchart of the development of the CNN model for component recognition.

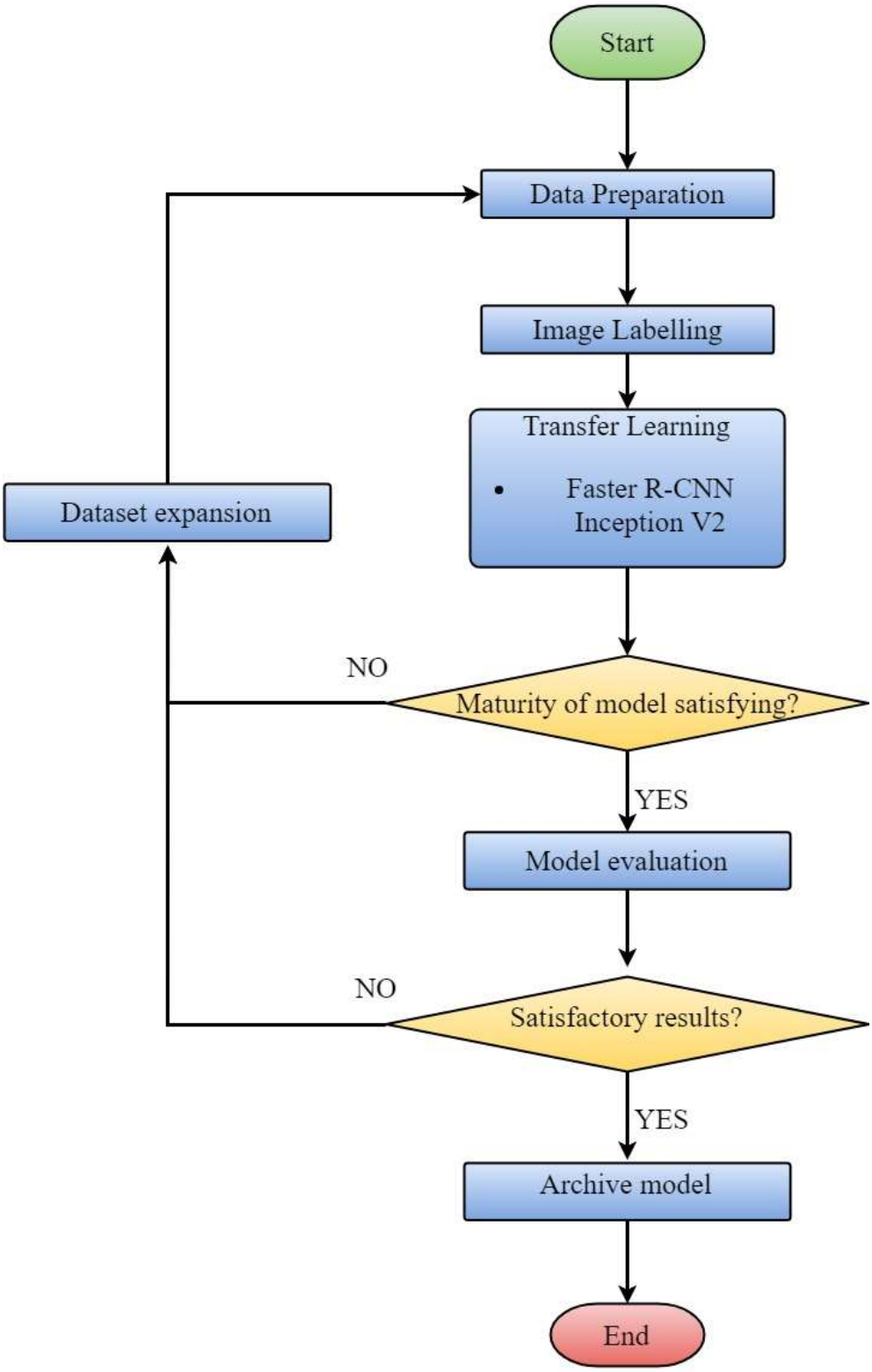


Figure 3.2: Overall flowchart of the development of the CNN model for defect localization.

3.2 PCB Component Classifier

PCB Component Classifier (PCB-CC) is the first part of the project which is to develop a system that can classify different components on the PCB. Basically, images of components are the input for PCB-CC and the output will be the prediction scores and prediction of component type.

3.2.1 Dataset Preparation

A total of 25 different components or classes are to be recognised. For each class, there is a total of 15 images. At first, the images are separated into 2 groups, where 5 are used as testing images and the remaining 10 images are augmented to increase the data size to 40. Then 80% of those 40 images, which are 32 images are used as training data while the other 20% are used as validation data. The training images are then fed into the CNN model for training the classifier. Validation of the model will also be performed by feeding the validation data to the classifier at every epoch of training to verify if the learning process of the CNN model is on the right track and know when the training has reached its peak performance. The test datasets will then be used to evaluate the performance of the whole system.

3.2.1.1. Data Collection

Since there are no publicly available datasets for PCB board, the dataset is built by requesting dummy PCB boards images from Vitrox. Dummy boards are used for tuning newly developed PCB inspection machines at Vitrox and they cover most of the defects that could arise in real life. The images of the PCB board are taken using Vitrox V510XXL AOI machine camera where they are under the same lighting condition, same scale and same angle. On the PCB board, there are 34 different components as shown in Table 3.1, each having different numbers of data. However, all these components cannot be selected because some of them do not have sufficient data such as “6X6 48LD” which only have 2 data, “20LD PLCC” which only have 6 data, and so on. For a decent training result to be achieved, it is decided to have at least 15 data for each component where 5 of the data are for testing while the rest are for training and validation. In Table 3.1, it is seen that the components with number of data greater than 15 are highlighted in green. Among all of them 25 different components are selected.

Table 3.1: Number of component's data collected (continue).

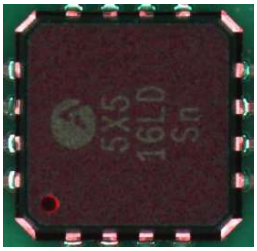
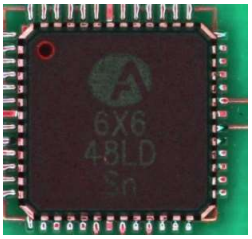
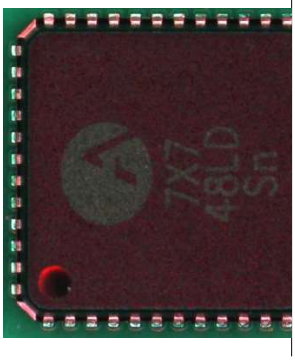
No.	Components	Number of data	Components which have number of data greater than 15
1	000B 	67	1
2	000 	38	2
3	5X5 16LDSn 	46	3
4	6X6 48LD 	2	
5	7X7 32LDSn 	52	4
6	7X7 48LDSn 	42	5

Table 3.1: Number of components' data collected (continue).

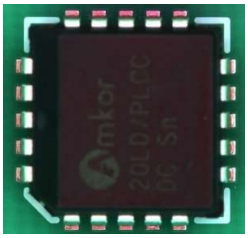
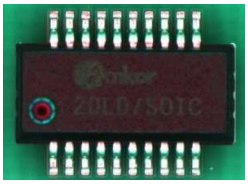
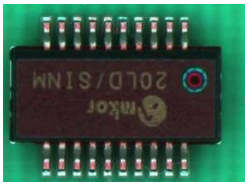
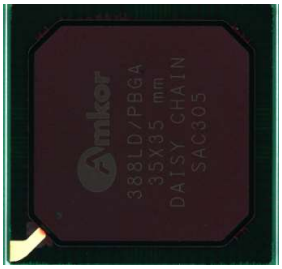
No.	Components	Number of data	Components which have number of data greater than 15
7	12X12 156LD 	24	6
8	20LD PLCC 	15	7
9	20LD S0IC 	6	
10	28LD S0IC 	27	8
11	20LD SINM 	16	9
12	68LD PLCC 	11	

Table 3.1: Number of components' data collected (continue).

No.	Components	Number of data	Components which have number of data greater than 15
13	131 	45	10
14	201 	73	11
15	202 	49	12
16	256LD PBGA 	15	13
17	272 	101	14
18	402B 	50	15
19	388LD PBGA 	13	
20	402 	66	16
21	7502 	33	17