Printed Circuit Board Defect Detection Using Convolutional Neural Networks

by

Ahmed Jawad Rashid (190021107)

Adiba Isfara (190021235)

Mohammad Aman Ullah (190021304)

Report on

EEE 4800 (Project and Thesis)

Department of Electrical and Electronic Engineering

Islamic University of Technology (IUT)

Gazipur, Bangladesh

June 2024

PRINTED CIRCUIT BOARD DEFECT DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Approved by:

Nadim Ahmed

Supervisor and Lecturer,

Department of Electrical and Electronic Engineering,
Islamic University of Technology (IUT),
Boardbazar, Gazipur-1704.

Date: …………………

Declaration of Authorship

This is to certify that the work presented in this thesis, titled **“Printed Circuit Board Defect Detection Using Convolutional Neural Network”**, is the result of diligent research conducted under the guidance of **Mr. Nadim Ahmed, Lecturer, Department of Electrical and Electronic Engineering (EEE), Islamic University of Technology (IUT)**. This work was entirely completed during our candidacy for a Bachelor of Science (B.Sc.) degree at this university. There has been no prior submission of any part of this thesis for any other qualification or degree. This thesis has meticulously acknowledged all primary sources of assistance and the published work of others. In instances where a work has been quoted, the sources have been comprehensively cited.

Ahmed Jawad Rashid (190021107)

Adiba Isfara (190021235)

Mohammad Aman Ullah (190021304)

Abstract

The increasing complexity and miniaturization of modern electronic devices necessitate highly reliable and defect-free Printed Circuit Boards (PCBs). Effective defect detection in PCBs is crucial to maintaining the quality and reliability of these devices. However, current PCB defect detection datasets exhibit significant limitations that hinder the development of robust and accurate models. Existing datasets are limited in scope, do not accurately mimic real-world defects, and fail to represent the diversity and complexity of industrial PCBs. For instance, PKU dataset is restricted to only a small amount of PCB boards and include defects introduced post-manufacturing using Photo Editing Applications, which do not closely mirror real-world manufacturing imperfections. Additionally, these datasets label each PCB board with only one fault class, despite real-life scenarios where multiple faults can occur simultaneously. These compounded limitations make the datasets less suitable for generalizing to the diverse and complex situations encountered in real-world PCB inspections. To address these challenges, this research aims to develop a comprehensive and realistic dataset created through chemical etching procedures, reflecting the true nature of manufacturing imperfections. Additionally, we trained advanced Convolutional Neural Network (CNN) models, including YOLOv8, HRNet, Cascade R-CNN, ATSS, RetinaNet, and Faster R-CNN, to detect six common PCB defects: missing pad, open circuit, short circuit, spur, spurious copper, and mouse bite. Our findings indicate that YOLOv8 demonstrated superior accuracy and speed, achieving a mean Average Precision (mAP) of 0.888 at 50% intersection over union (mAP50) and a detection speed of 16.3 frames per second (FPS). HRNet, while achieving the highest mAP50 of 0.905, was less suitable for real-time applications due to its lower frame rate of 9.2 FPS. ATSS and Cascade R-CNN offered balanced performance with mAP50s of 0.88 and detection speeds of 14.51 FPS. In contrast, RetinaNet and Faster R-CNN were less effective due to lower accuracy and slower processing times. This study underscores the inadequacies of existing PCB datasets and the necessity for more accurate and efficient defect detection models.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Acronyms

Acronym	Full Form
AOI	Automated Optical Inspection
AP	Average Precision
ATSS	Adaptive Training Sample Selection
BN	Batch Normalization
CAD	Computer-Aided Design
CA	Controlled Autoregressive
CNN	Convolutional Neural Network
CSPDarknet	Cross Stage Partial Darknet
CV	Computer Vision
DCR-YOLO	Dual-Channel Residual YOLO
DRC	Design Rule Check
EMA	Exponential Moving Average
FN	False Negatives
FP	False Positives
FPN	Feature Pyramid Network
FPS	Frames Per Second
HRNet	High-Resolution Network
IoU	Intersection over Union
mAP	Mean Average Precision
MBBi	Multi-Scale Feature Extraction and Balanced Beam Intersection
MLP	Multi-Layer Perceptron
MSD-YOLO	Multi-Scale Detection YOLO
NAS FPN	Neural Architecture Search Feature Pyramid Network
NMS	Non-Maximum Suppression
PCB	Printed Circuit Board

RCNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RoI	Region of Interest
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SLP	Single-Layer Perceptron
SPICE	Simulation Program with Integrated Circuit Emphasis
SPP	Spatial Pyramid Pooling
SSD	Single Shot MultiBox Detector
SURF	Speeded-Up Robust Features
SVG	Scalable Vector Graphics
TN	True Negatives
TP	True Positives
YOLO	You Only Look Once
YOLOXPAFPN	YOLO eXtended Path Aggregation Feature Pyramid Network

Thesis Framework

Chapter 1: Introduction

This chapter provides an overview and introduction to the research topic, outlining the goals and motivations of the project.

Chapter 2: Background and Related Work

This chapter covers fundamental concepts of Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs), followed by a comprehensive review of literature on PCB defect detection using CNNs.

Chapter 3: Methodology

Detailed methodology of PCB manufacturing using chemical etching, data processing methods including annotation and pre-processing, various model architectures utilized, and evaluation metrics employed.

Chapter 4: Results and Discussion

Presentation and discussion of performance metrics such as mAP curves, loss curves, addressing class imbalance, and qualitative analysis of different baseline architectures.

Chapter 5: Conclusion

Summary of findings from the research, implications of the results, and future research directions for further advancements in PCB defect detection using CNNs.

Chapter 6: OBE Requirements

Outcomes-Based Education (OBE) requirements are discussed in this chapter, focusing on how the research aligns with educational objectives and outcomes.

References

A comprehensive list of all the sources cited throughout the thesis, providing credibility and further reading for interested readers.

Chapter 1

Introduction

1.1 Overview

A Printed Circuit Board (PCB) serves as the foundational component in virtually all electronic devices, providing both the physical structure and electrical connectivity necessary for the operation of electronic circuits [1]. PCBs consist of a flat sheet of insulating material, typically fiberglass, onto which conductive pathways are etched or printed to connect various electronic components. Despite their critical role, PCBs are susceptible to numerous defects during the manufacturing process. These defects can arise from inaccuracies in etching, soldering, or component placement, leading to issues such as open circuits, short circuits, missing pads, spurs, spurious copper, and mouse bites [2]. Such defects can compromise the functionality, reliability, and longevity of electronic products, making effective defect detection essential.

In contemporary manufacturing industries, ensuring product quality detection within industrial environments stands as a critical imperative. Despite this, many factories continue to rely predominantly on traditional manual detection methods. As PCBs advance towards higher precision, complexity, and performance standards, manual inspection struggles to adequately meet the demands of quality assurance. Consequently, there is a growing inclination towards leveraging computer vision as a technological solution to enhance inspection efficiency. In pursuit of this objective, the focus of PCB defect detection shifts towards employing deep-learning frameworks, prominently Convolutional Neural Networks (CNNs). These frameworks are adept at analyzing images, possessing the capability to discern intricate patterns within visual data. CNNs have thus emerged as formidable tools for automating and improving the accuracy of defect detection processes previously reliant on human operators [3]. Through the adoption of CNNs in quality inspection, industries aim to streamline operations, reduce error margins, and elevate overall production standards. This paradigm shift underscores a significant stride towards integrating advanced technological solutions in manufacturing, marking a departure from conventional practices towards more efficient and reliable quality control mechanisms. This research aims to leverage advanced CNN architectures, such as YOLO, Faster R-CNN, and RetinaNet, to enhance defect detection in PCBs. The study will provide a comprehensive approach to dataset development, model training, and optimization, ultimately aiming to improve quality control in PCB manufacturing.

1.2 Motivation and Scope

The increasing complexity and miniaturization of modern electronic devices necessitate highly reliable and defect-free Printed Circuit Boards (PCBs). However, current datasets used for PCB defect detection, such as those created by Peking University and DeepPCB, exhibit significant limitations. The Peking University dataset is restricted to only ten PCB boards, which limits its representativeness of the diverse range of PCBs used in the industry [4]. Additionally, the artificially introduced defects in this dataset do not accurately mirror real-world manufacturing imperfections, reducing its utility for training models aimed at realistic industrial applications. The labeling of each PCB board with only one fault class further detracts from its relevance, as PCBs in actual manufacturing environments often contain multiple simultaneous faults [4].

Similarly, the DeepPCB dataset is hampered by its use of cropped images from a single PCB, failing to capture the comprehensive variety of layouts and defects present in real-world scenarios [5]. The absence of full-dimensional depth in PCB design schematics and the restriction to black-and-white images further limit the dataset’s applicability, as these simplifications do not reflect the complexities and color variations inherent in actual PCB inspections.

Given these constraints, there is a clear need for a more comprehensive and realistic dataset that can better support the development of effective PCB defect detection models. This research aims to address this gap by creating our own dataset that overcomes the shortcomings of existing datasets. By incorporating a wide range of real-world PCB defects and scenarios, our dataset will facilitate the training of robust and accurate Convolutional Neural Network (CNN) models.

The integration of deep learning and computer vision techniques has already demonstrated significant improvements in PCB defect detection. State-of-the-art models such as YOLOv8, YOLO-MBBi, and Transformer-YOLO have shown substantial advancements in detection accuracy and efficiency [6], [7], [8]. Moreover, the development of lightweight models like MSD-YOLO and DCR-YOLO underscores the industry's need for real-time solutions. By extending the use of hybrid models and incorporating contextual data, our research aims to further enhance the defect detection process, contributing to improved quality control in PCB manufacturing.

In summary, the motivation for this research is driven by the inadequacies of existing PCB datasets and the necessity for more accurate and efficient defect detection models. By creating a comprehensive dataset and leveraging advanced CNN architectures, we aim to significantly advance the state of PCB defect detection, ultimately contributing to higher quality and reliability in electronic devices.

1.3 Problem Statement

The primary problem addressed by this research is the inadequacy of current PCB defect detection datasets and methodologies to meet the demands of real-world industrial applications. This inadequacy leads to inefficiencies and inaccuracies in defect detection, which can compromise the quality and reliability of electronic products. To address this problem, our research aims to develop a new, comprehensive dataset that accurately represents the diversity and complexity of real-world PCBs and their defects. Furthermore, we aim to train and optimize advanced CNN models to achieve high accuracy, sensitivity, and precision in detecting six common types of PCB defects: missing pad, open circuit, short circuit, spur, spurious copper, and mouse bite.

Existing datasets, such as those developed by Peking University and DeepPCB, exhibit several critical drawbacks. The Peking University dataset is limited to only ten PCB boards, which does not adequately represent the diversity of PCBs produced in the industry. Furthermore, the defects in this dataset are artificially introduced using applications like MS Paint, which fail to accurately mimic real-world manufacturing imperfections [4]. This artificial nature of the defects reduces the dataset’s applicability for training models intended for practical industrial use. Additionally, the dataset’s labeling of each PCB board with only one fault class does not reflect the reality of manufacturing environments where multiple faults can occur simultaneously.

Similarly, the DeepPCB dataset is restricted to cropped images of a single PCB, which limits its ability to capture the full range of layouts and defects present in actual PCB production. The lack of full-dimensional depth in PCB design schematics and the restriction to black-and-white images further limit the dataset’s effectiveness. These simplifications fail to account for the color variations and complexities inherent in real-world PCB inspections, reducing the model’s generalizability and accuracy.

Given these constraints, there is a pressing need for a more comprehensive and realistic dataset that can effectively support the development of robust PCB defect detection models.

Additionally, there is a need for advanced Convolutional Neural Network (CNN) models that can accurately detect and classify multiple types of PCB defects with high sensitivity and precision.

1.4 Research Challenges

One of the primary challenges in PCB manufacturing is the inaccurate etching of traces during the etching process. This inaccuracy can result in either excessively wide copper traces or excessively narrow traces. Wide traces lead to the presence of unwanted conductive materials, unnecessary holes, and open circuits, potentially causing short circuits. Conversely, narrow traces and gaps due to excessive etching may lead to open circuits. These inaccuracies are classified as fatal defects because they directly cause PCB malfunctions.

The etching process is influenced by various parameters such as material removal rate, etch factor, undercut, concentration of etchant, etching time, and etchant temperature [9]. Understanding the combined effects of these parameters is crucial for optimizing the etching process. Optimal parameter combinations for maximum material removal rate and undercut within selected control parameters must be obtained using analysis of variance. Significant parameters affecting undercut include temperature and time, with temperature being particularly critical. Non-uniform acid circulation, gas bubbling during etching, and mask resolution limitations further complicate achieving consistent etching quality.

During the etching process, the edges of copper traces are neither completely smooth nor perfectly vertical. Rough edges, known as edge definition issues, occur due to mask resolution limitations, non-uniform acid circulation, and gas bubbling [10]. Additionally, as acid etches exposed copper, a sidewall forms, which is also attacked by the acid, leading to undercutting. This phenomenon, known as etch back, further complicates maintaining precise trace dimensions and integrity.

PCBs can exhibit a wide variety of defects, each with distinct visual characteristics. Developing a model that can accurately identify and classify multiple defect types presents a significant challenge. The model must be robust enough to handle the variability in defect appearances and accurately differentiate between various fault types such as missing pads, open circuits, short circuits, spurs, spurious copper, and mouse bites.

In summary, the challenges of this research encompass addressing the inaccuracies in the etching process, understanding the combined effects of various etching parameters, managing edge definition and undercutting issues, and developing a model capable of accurately identifying diverse defect types. Overcoming these challenges is essential for improving PCB defect detection and ensuring the production of high-quality, reliable PCBs.

1.5 Research Contributions

This research makes significant contributions to the field of PCB defect detection and aims to address several critical gaps in current methodologies. The primary contributions of this study are threefold:

1. **Development of a Comprehensive PCB Defect Dataset:** The first major contribution is the creation of a novel dataset specifically designed to overcome the limitations of existing PCB datasets. Current datasets, such as those developed by Peking University and DeepPCB, suffer from significant shortcomings, including limited representation of PCB varieties, artificial defect introduction, and lack of complexity in defect representation. Our dataset will include a diverse array of PCBs with real-world defects, accurately reflecting the range of issues encountered in actual manufacturing environments. By incorporating multiple fault types and realistic defect scenarios, this dataset will provide a robust foundation for training and evaluating advanced defect detection models.
2. **Training and Optimization of Advanced CNN Models:** The second contribution focuses on the training and optimization of state-of-the-art Convolutional Neural Network (CNN) models to detect PCB defects with high accuracy, sensitivity, and precision. We will employ a variety of advanced CNN architectures, including YOLO (You Only Look Once), Faster R-CNN, RetinaNet, HRNet, Cascade R-CNN, and ATSS. These models will be meticulously trained and fine-tuned to enhance their performance in identifying PCB defects. The optimization process will ensure that these models can effectively and efficiently detect six common types of PCB defects: missing pad, open circuit, short circuit, spur, spurious copper, and mouse bite.

In summary, this research contributes to the development of a new, comprehensive PCB defect dataset, the training and optimization of advanced CNN models for high-precision fault detection, and the enhancement of both academic knowledge and practical applications in PCB defect detection and electronics manufacturing.

Chapter 2

Background and Related Work

This chapter delves into the foundational concepts and significant advancements in the field of computer vision, particularly focusing on neural networks and their applications in PCB defect detection. Computer vision, a subfield of artificial intelligence (AI), enables machines to interpret and understand visual information from the world. It involves the development of algorithms and models that can process, analyze, and make sense of image and video data, which is crucial for tasks such as object detection, classification, and recognition [11].

2.1 Artificial Neural Networks

2.1.1 Basic Structure of ANNs

Axons and dendrites are the ways or the paths through which the neurons or nerve cells are linked in the human brain. The dendrites are used by neurons to intake signals from sensory organs or the external world while axons are used to pass signals from one neuron to another [12]. It allows neurons to send out messages and to perform computations across the neural network.

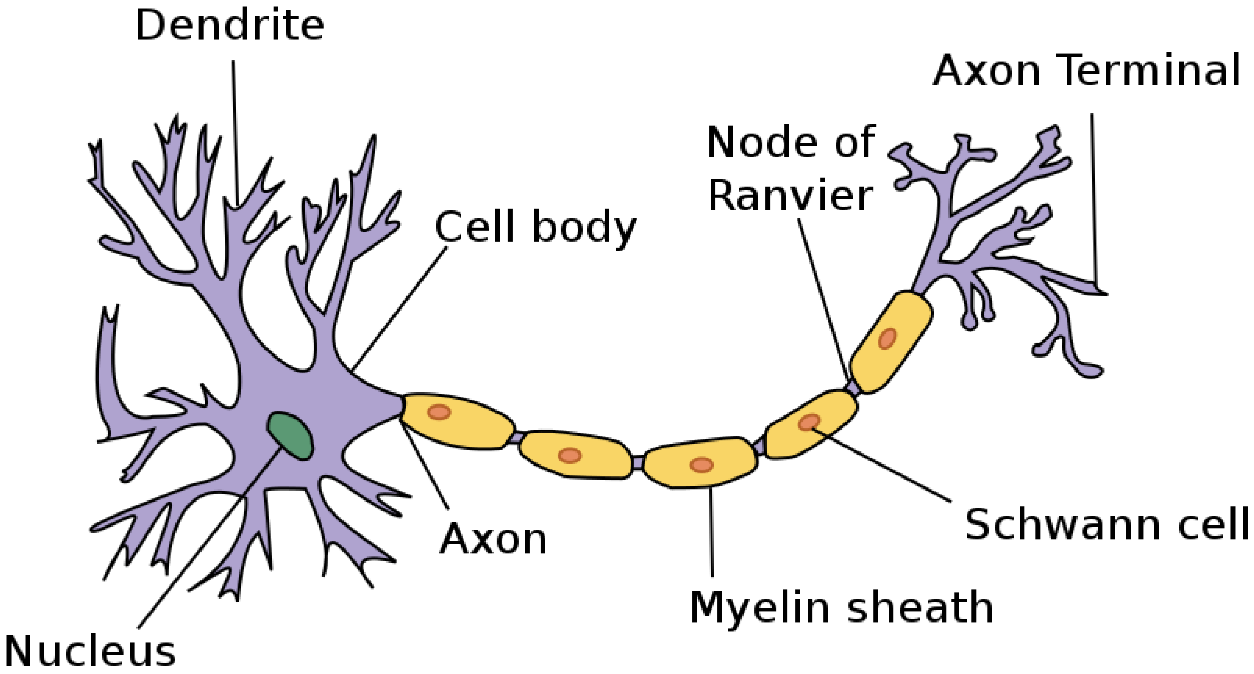


Figure 1 Neuron Structure [13]

ANNs are derived from nodes that mimic real neurons in the human brain. These are the nodes and they are linked with each other and they can communicate with each other. In ANNs, the input data is taken by nodes which in turn perform some computations on the data using the weights assigned to each connection [13]. The results are then transmitted to other neurons. The activation or node value of each node is the output produced by that node.

The learning process in ANNs comprises the modification of weights assigned to each link connecting two neurons. This change enables the network to improve on the performance of tasks like pattern recognition or decision making. An ANN can enhance the proficiency of data processing and analysis by adjusting these weight values based on the prediction errors, which is the same way the human brain acquires adaptive learning.

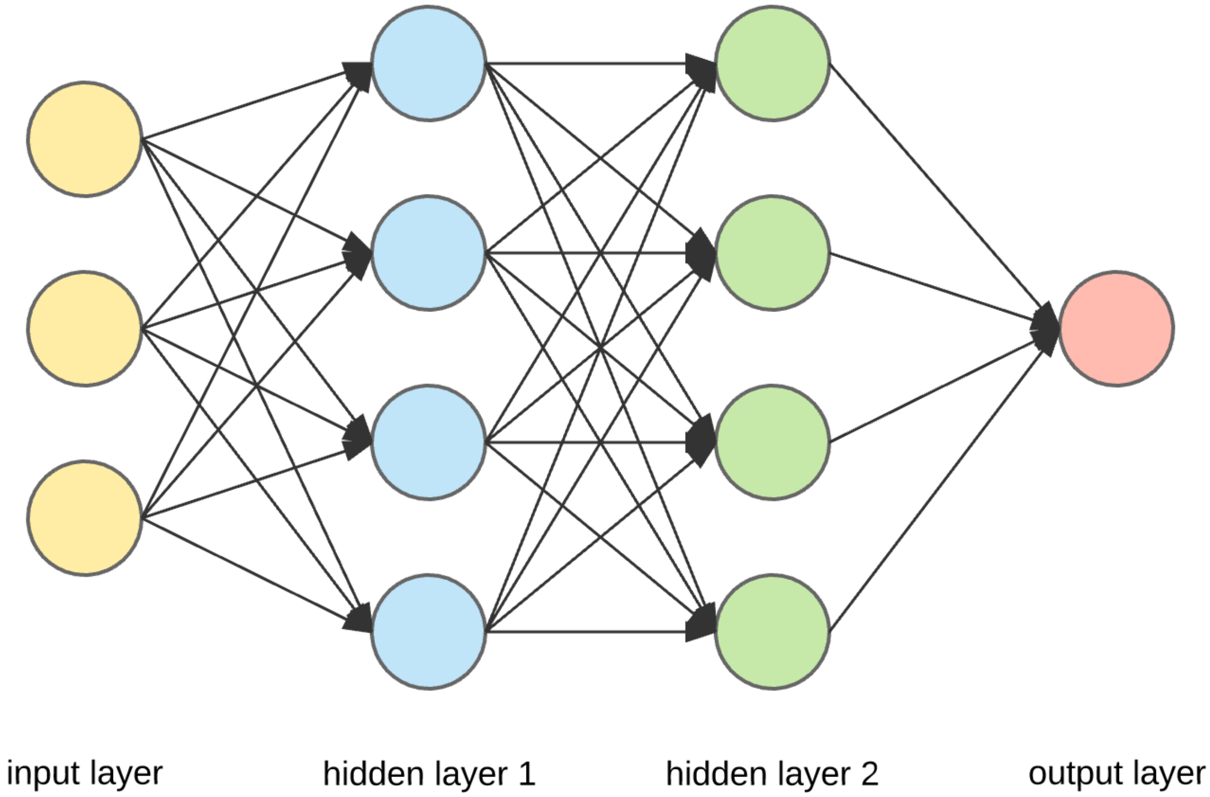


Figure 2 ANN Overview [14]

2.1.2 Types of Artificial Neural Networks

There are two types of Artificial Neural Networks: FeedForward and Recurrent.

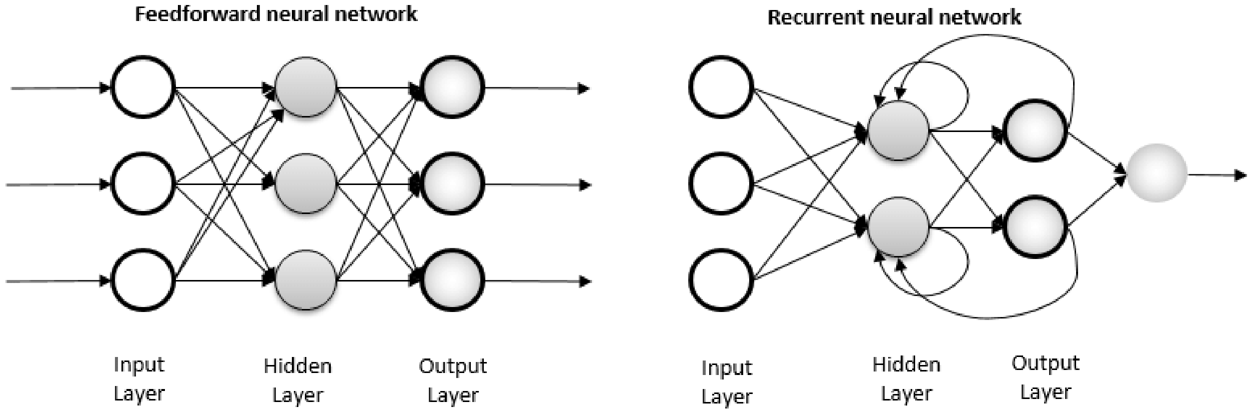


Figure 3 Types of Artificial Neural Networks [15]

2.1.3 Feed Forward ANN

Unlike other ANNs, this one is a feed-forward ANN, and the information flow is one way only. They are used in relaying information or communication from one unit to the other. They are devoid of feedback mechanisms. Some of the uses of ANN are in pattern formation, recognition, and categorization. They contain a finite number of inputs and outputs.

Single Layer Perceptrons (SLPs): The single-layer perceptron (SLP) is the basic building block of artificial neural networks (ANNs) and is characterized by a simple structure and functions. Structurally, an SLP is a model consisting of one output neuron layer connected directly to the input features [16]. It does not have hidden layers which would make computations more complex. This simplicity is also observed in the activation function, which usually is a step-function. This function gives an output of 1 if the sum of weighted inputs is greater than a threshold value; otherwise, the output is 0.

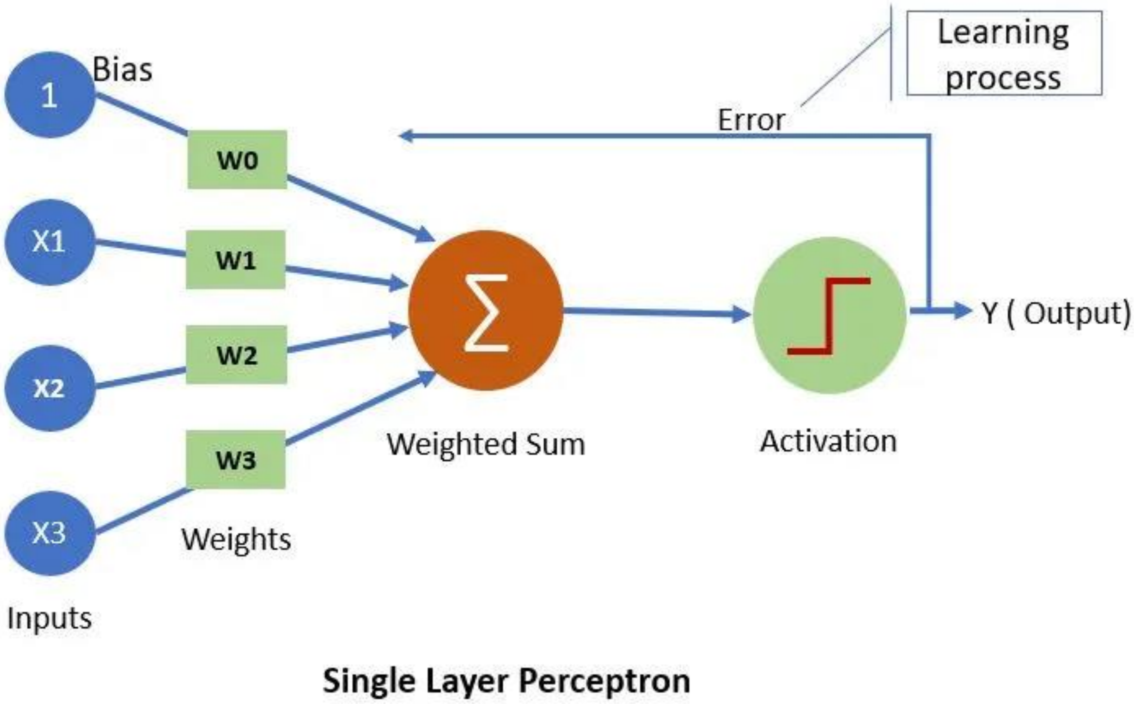


Figure 4 Single Layer Perceptron [17]

The perceptron learning algorithm is credited for training the SLP through cycling through the weights assigned to inputs in order to account for a mismatch between the predicted and actual outputs. The rationale of this process of making adjustments is to minimize the classification error, which enhances the ability of the model to recognize features in the training data.

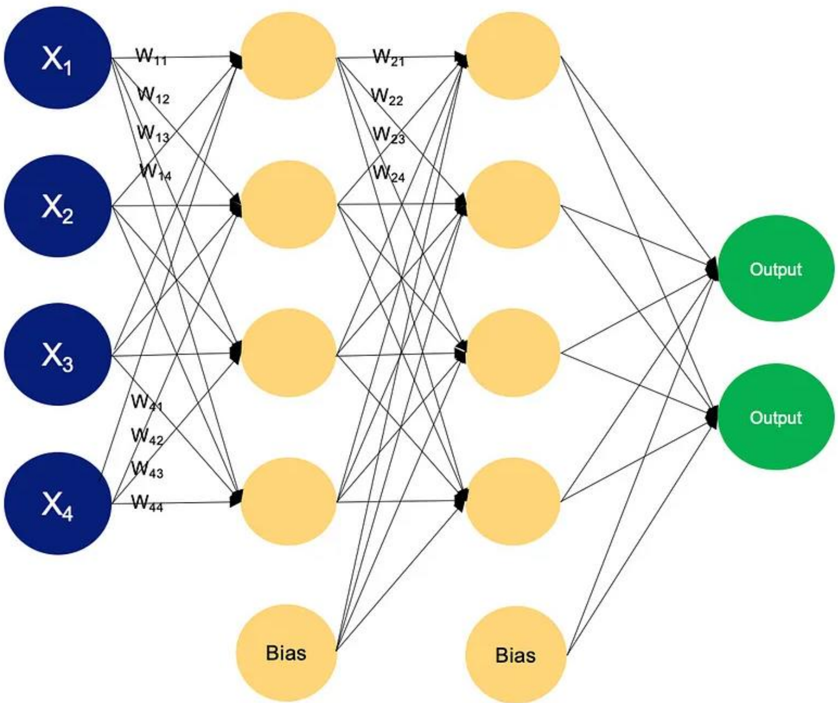


Figure 5 Multi-Layer Perceptron

However, the SLP is limited to the extent that it can only handle linearly separable patterns because of the linear structure and the requirement of linear separability. This applicability is limited to cases where the data elements can be separated using a straight line or hyperplane in the input space. The SLP is mainly applied in tasks that require the identification of the presence or absence of a phenomenon, the identification of patterns and trends, and linear regression.

Multi-Layer Perceptrons (MLPs): The use of MLPs was a great improvement over the earlier single-layer perceptron models. They were also able to model and learn complex relationships within given data sets since MLPs consisted of multiple layers of interconnected neurons.

These architectures are fundamentally different from their predecessors, such as the perceptron, and consist of an input layer, one or more hidden layers, and an output layer. It has the ability to learn complex features from input data because of the weighted connections between layers, which are made of several neurons [18].

In contrast to the SLP, the MLP represents a significant improvement, as it offers a number of important enhancements that extend the model's capabilities and utility across various fields. The use of multiple hidden layers is the key to its advantage, because it gives the MLP ability to discover complex patterns and dependencies in the data. It does this by using its hidden layers which are not limited to the linear separability of the function. Since the mappings between inputs and outputs are intricate and made possible by these functions, the model is able to capture a wider spectrum of data complexity.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs, or ConvNets) are described as a class of deep neural networks designed to process data with a grid-like topology, such as images [19]. It has been very useful most in computer vision where problems such as image classification, object detection, and even face recognition are solved. Based on the convolution layer, pooling layer and fully connected layers, CNNs are supposed to self-construct the spatial pyramid of features with gradually increasing complexity during the back propagation. These networks rely on local connectivity within the neurons to learn from and interact with the higher dimensional annotations of the images. In contrast to most other neural network structures where feature extraction is usually required, CNNs are capable of recognizing features from the raw image data without prior feature extraction. Due to this capability, CNNs are highly efficient and widely used for various sorts of visual recognition tasks.

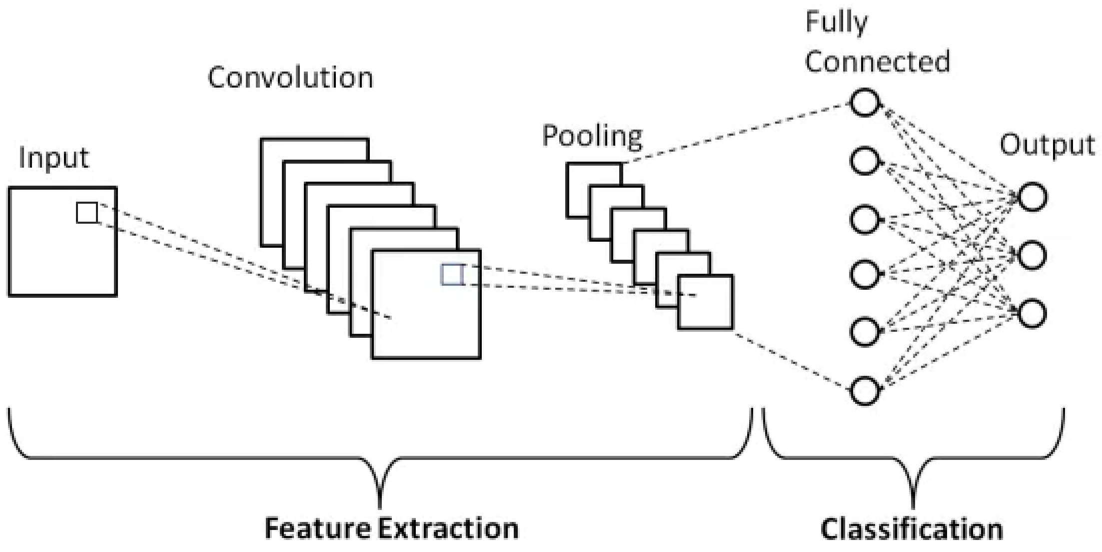


Figure 6 Basic architecture of a CNN [20]

2.2.1 Convolutional Layers

The convolutional layer is a basic component of a Convolutional Neural Network (CNN) and is the building block that the other layers are based on. Basically, it is expected to automatically and adaptively learn spatial hierarchies of generated features from input images. Each layer of a convolutional layer uses a set of kernels on the image by rastering over the parts of the image and performing a mathematical operation called convolution [19]. This operation produces a matrix of feature maps that may encode any possible features of an image, be it edges, texture, or pattern.

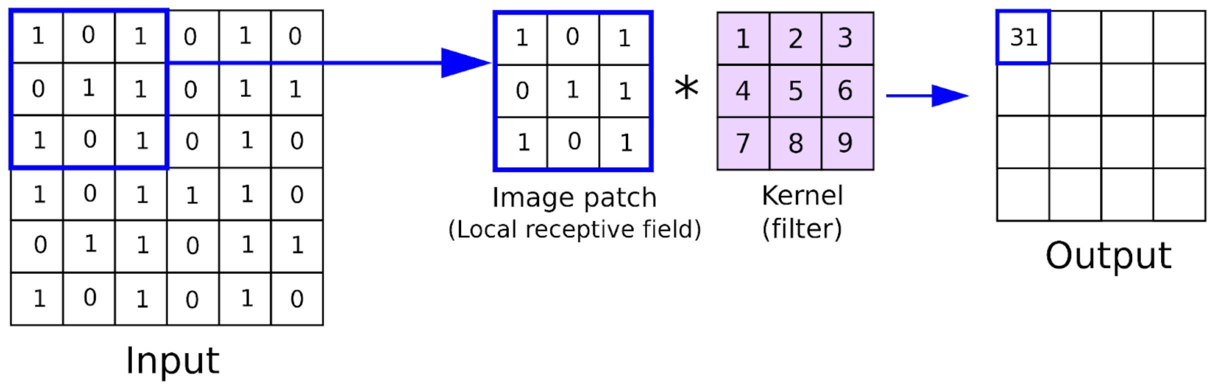


Figure 7 Convolution operation [21]

One of the interesting characteristics of convolutional layers is that they use the filter on the whole image without the need to adjust it for each of the patches, as is the case with other methods. The outcome of the convolution operation is passed through a non-linear activation function, which is commonly ReLU, which introduces non-linearity into the model and hence allows the model to learn non-linear and complicated features. The operation has three hyperparameters which are the filter size, the stride which is used to move the filters, and padding which determines the handling of the edges of the images.

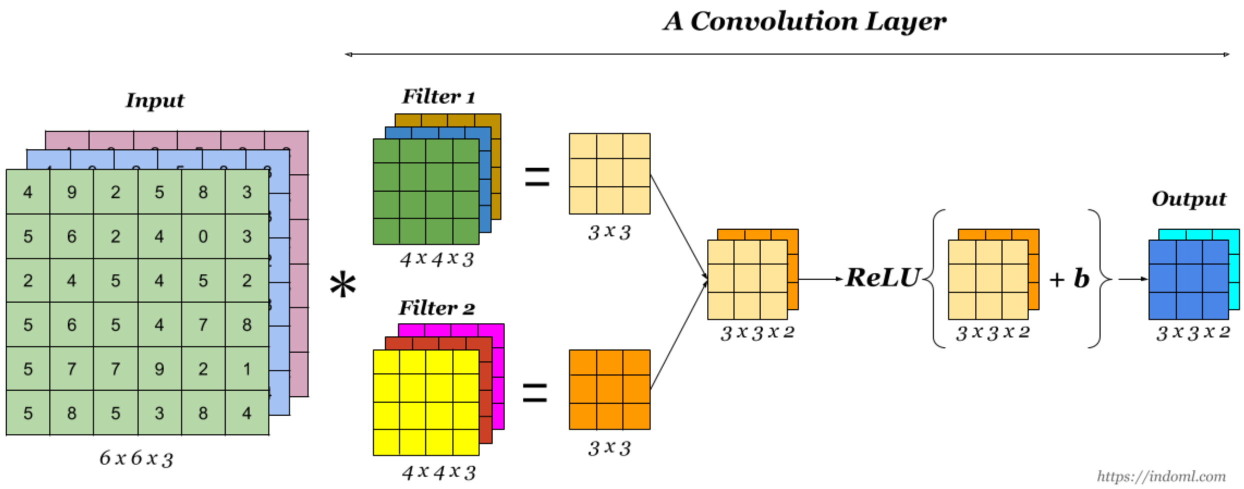


Figure 8 A Convolution Layer [22]

Given an input with dimensions W x W x D and Dout kernels, each with a spatial size F, stride S, and padding P, the output volume size can be calculated using the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

The resulting output volume will have dimensions Wout x Wout x Dout.

2.2.2 Filter Size

Kernel size, also known as filter size in convolutional neural networks, is another tunable parameter that defines the dimension of the filter matrix used in the convolution process. Filters are small matrices of the same size as the image, for example 3×3 or 5×5, that are displaced on the input image to recognize some features. The size of the filter determines the receptive field of the convolutional layer, which is the portion of the input image that the filter processes at a particular moment in time [19].

3×3 is used most often because of the high level of detail in the input data, and it is possible to construct even deeper architectures by adding more layers. Although filters of a larger size, such as 5 x 5 or 7 x 7, contain information from a larger field of context in one layer, they are used less frequently due to the increased computational complexity and can be effectively used when transitioning to deeper networks. This defines the level of detail of the features that the network is capable of distinguishing; filters of small sizes are fine and select very detailed features while those of large sizes get more general features.

$$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}\qquad \frac{1}{273}\begin{bmatrix}1&4&7&4&1\\4&16&26&16&4\\7&26&41&26&7\\4&16&26&16&4\\1&4&7&4&1\end{bmatrix}\qquad \frac{1}{1003}\begin{bmatrix}0&0&1&2&1&0&0\\0&3&13&22&13&3&0\\1&13&59&97&59&13&1\\2&22&97&159&97&22&2\\1&13&59&97&59&13&1\\0&3&13&22&13&3&0\\0&0&1&2&1&0&0\end{bmatrix}$$

Figure 9 3x3, 5x5, and 7x7 kernels

2.2.3 Stride

Another hyperparameter in CNNs is stride which represents a certain step that is used to demonstrate how the filter moves across the input image during the convolution process. Stride defines the level by which the filter shifts from one step to another in terms of the horizontal and the vertical axis. The most selected value of stride is 1, as it moves the filter with one pixel at a time, having more dense and interlaced ways to operate on the input image [23].

Stride values of 2 or more indicate that the filter moves forward by more pixels at a time. The convolution operation thus introduces a reduction in the size of the output feature map in the spatial dimension, which effectively means down-sampling of the input. When the stride is large, issues like the time taken to perform a computation or the number of parameters can be handled, but there are details on the input image that will be missed. Therefore, the traversal type determines the relationship between the time taken and the accuracy of the model to capture minute details of a dataset.

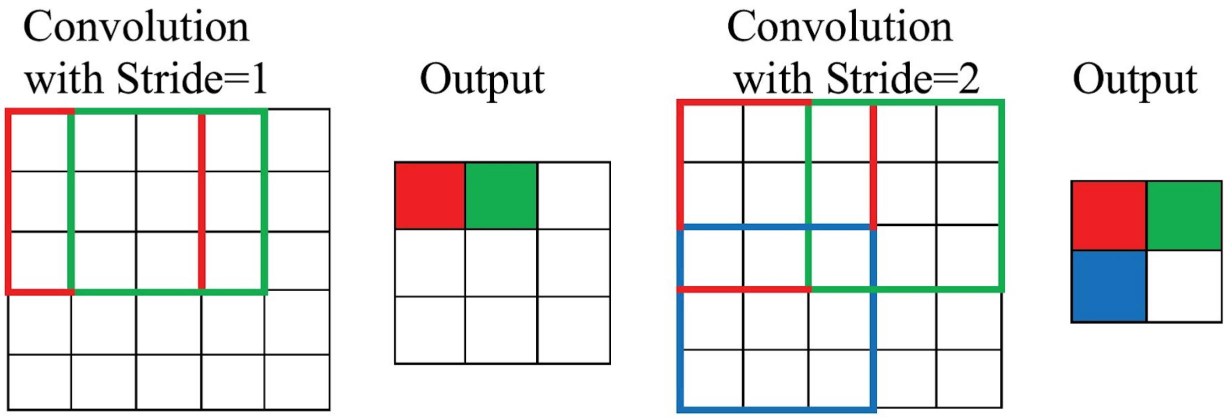


Figure 10 Convolution with Strides

2.2.4 Padding

Padding is generally applied in CNNs to control the spatial size of the feature map. However, if a filter is put on top of an input image, then the nature of the output feature map may be less than the size of the input depending on the size and stride of the filter. Padding is the act of adding some pixels outside the outer rim of the input image to make sure that the filter has enough space to surround the image edges [19]. It is required to preserve the spatial dimensions of the image and to avoid losing useful features at the borders of the image during convolution. It also helps in remembering the output size which is very beneficial when creating many layers of networks without having the feature maps to become very small [24].

Given an activation map with dimensions W x W x D, a pooling kernel of spatial size F, and stride S, the output volume size can be calculated using the following formula:

$$W_{out} = \frac{W - F}{S} + 1$$

The resulting output volume will have dimensions of Wout x Wout x D.

There are three main types of padding:There are three main types of padding:

Valid Padding (No Padding): All padding steps are excluded in this method and the filter is applied only in the sub-image in which the filter fits completely. This is usually coupled with a decrease in the size of the feature map output as it takes in an input.

Same Padding: This method adds zeros around the boundary of the image for the output feature map to have the same dimension as the input. Same padding also enables the completion of the filter across the edges of the images to ensure that it is a complete process.

Full Padding: This method also incorporates additional margins more than necessary to make the filter transfer, as many elements of the input may extend even past the borders of the image. Therefore, with full padding, the feature map is larger than the input layer on which full padding is used.

2.2.5 Pooling Layers

A pooling layer is one of the main components of the CNN that can help to reach a certain degree of downsampling. Such a reduction helps in mitigating the computation load and storage required, as well as making a certain level of invariance. Pooling does so in a manner that averages the total number of features present in patches of the feature map which is relevant information while ignoring other less relevant details.

There are two types of pooling layers that are widely used and they are the max pooling layer and the average pooling layer. Max pooling is the process of selecting the maximum datum in a special district in the set feature map to acquire the dominant aspect of this area. In smooth pooling, average pooling assesses the average of the values in the region of the feature map as described below. The most used pooling is the max pooling since it is normally capable of identifying the features highlighted by the convolutional layers.

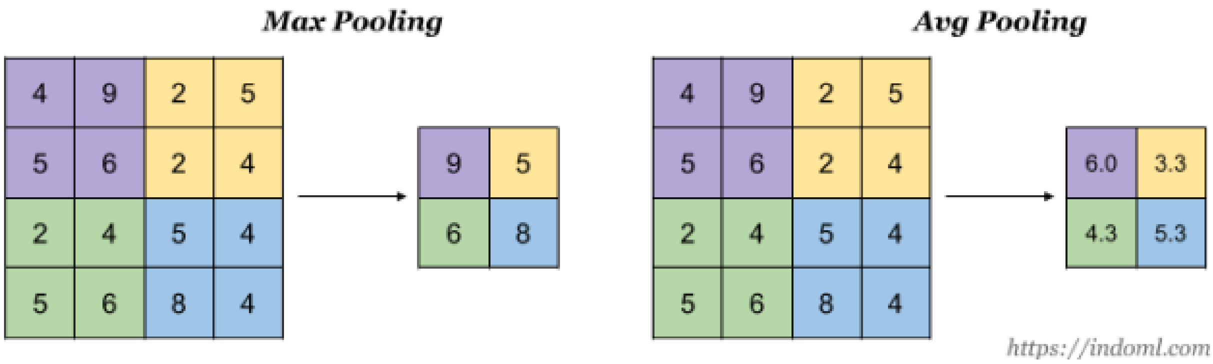


Figure 11 Pooling

Pooling layers help in cutting down on the parameters into a reduced quantity and avoid cases of overfitting. Through the process of summarizing features from regions, pooling layers contributing to the network enable the network to be less sensitive to variations and distortions in the input images. This is useful in instances such as object detection, where objects viewed from a top-bottom view may appear in different scales or positions within the image.

2.2.6 Fully Connected Layers

There are other components of Convoluted Neural Networks known as dense layers or fully connected layers which are usually incorporated at a later stage of the network. These layers are used to calculate the final outcome in the form of a decision or a prediction that is made about the object that was detected and/or recognized from the features derived from the subsequent convolutional and pooling layers. Fully connected layer means that each neuron from a layer is connected with each neuron in the previous layer so as to allow the network to combine all the learned features.

These are fully connected with the feature maps which are in one dimensional and stacked layers format as received from the convolution and pooling layers. This transformation allows the network to gather back the features that had been spatially distributed across the forming net and form an overall picture of the input image [23]. Fully connected layers necessitate backpropagation, which means that the target output is passed backwards in order to reduce the error for each layer in the network. This process allows for the modification of the weights and bias in such a manner that minimizes the error hence improving the prediction.

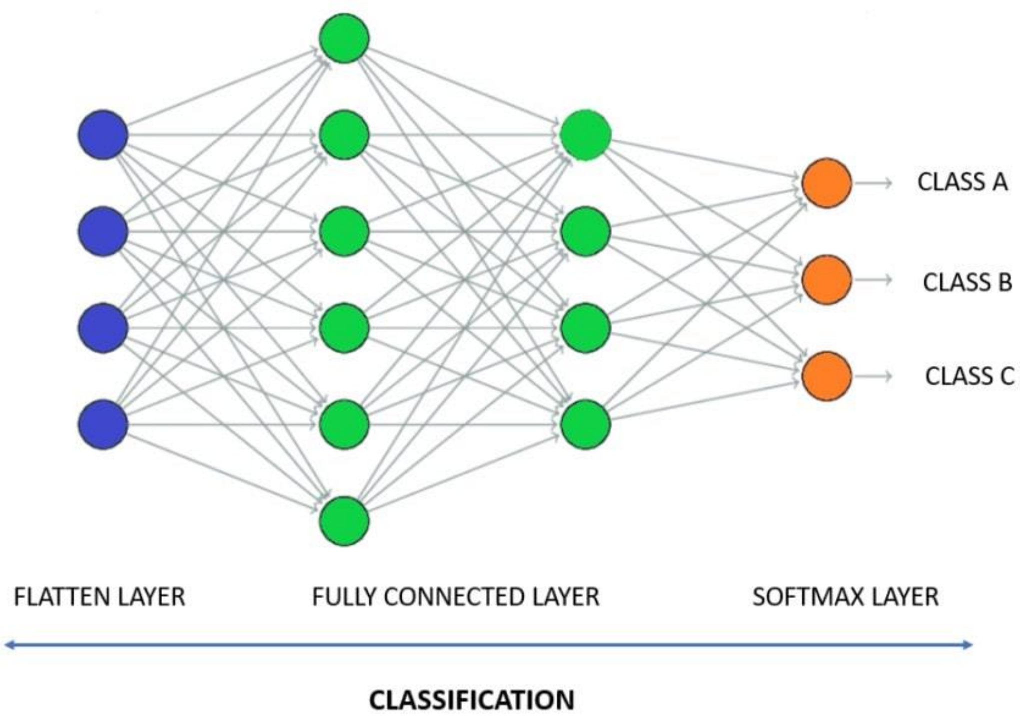


Figure 12 Classification of Convolutional Layers [25]

2.3 Vision Task

Computer vision functionality is the use of artificial intelligence to identify the right materials from images or videos and extract relevant information. These tasks turn out to be very useful, as they assist machines in perceiving the objects and then decide on the further course of action. Most of the contemporary vision applications can be attributed to this category and they include classification, object detection, semantic segmentation, and image generation [26]. Due to the development of these algorithms, particularly the CNNs, this technology has become a reality and high accuracy and efficiency could be achievable in areas such as health care, self-driving cars, and surveillance.

2.3.1 Object Detection

Image classification and object detection are two key sub-tasks of computer vision, a domain that aims at training computers to understand visual information from the environment. In image classification, the primary goal is to distinguish the existence of an object in the image, for instance, an image of a cat or an image of a dog. This task is associated with identification of unique features which distinguish one category from another, for instance, pointed ears of house cats compared to rounded ears of Golden Retrievers or lighter colored nose of house cats compared to that of Golden Retrievers. In the past, feature engineers would have to hand-code

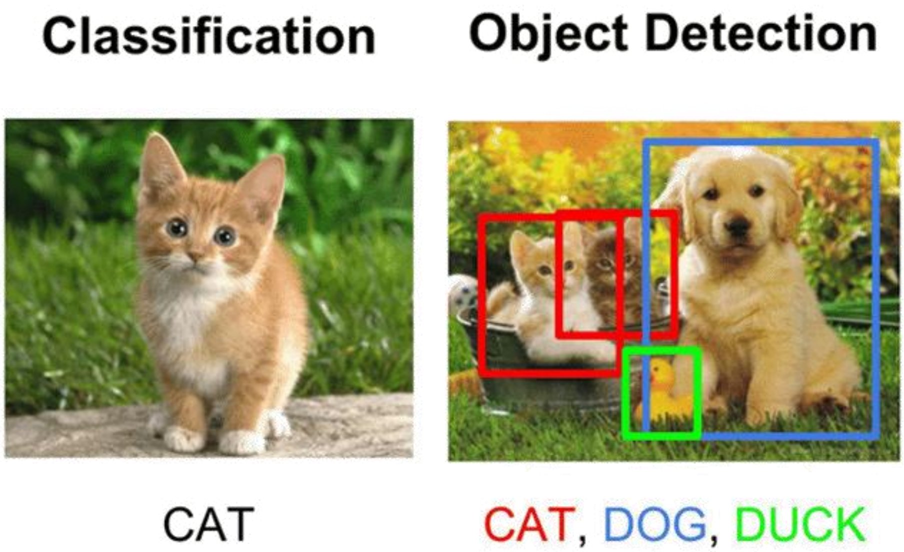


Figure 13 Classification vs Object Detection

the algorithms to extract these features, a process that was both time-consuming and challenging. However, the recent introduction of neural networks in the late 1990s changed this approach. In contrast, in neural networks, feature extraction is not a separate step as the model learns to perform feature extraction during the training phase. This means that the feature extraction that is done automatically is not only the human-understandable features but also numerous other patterns that greatly improve the classification accuracy.

Object detection goes a step further from image classification not only does it identify if the objects are present in the image but also the position of the objects. This process includes region proposal, whereby the algorithm searches the image for areas that may contain areas of interest, highlighting regions that are likely to contain objects. The proposed regions are then passed through image classifiers which are trained to identify certain objects of interest [27]. For instance, in the context of an image depicting a cowboy astride a horse, the object detector will generate regions and classify them to show that there is a person, a horse, and a hat, but at the same time, it will correctly localize regions that do not contain any objects.

The main difference between image classification and object detection is that image classification is a basic building block of many applications in computer vision, ranging from automobiles that are self-driven to detect and avoid objects to security cameras that detect and track intruders [28]. These technologies have advanced the study of computer vision, as they allow for the development of better and more accurate systems for interpreting visual data. Neural networks have been specifically incorporated to improve the field, making it possible to create stronger and flexible algorithms that can accommodate the richness of images that is evident in the real world. Semantic segmentation entails the partitioning of an image into regions of different objects that are present in the image [28]. This process identifies what is in the image such as a cat, a dog, or car without identifying where in the image they are. However, object localization is more advanced than object recognition since it not only categorizes the objects but also identifies the location of the objects. Localization entails placing rectangles around each identified object, which gives specific spatial information on the objects in the picture. Combined, these tasks are essential in applications where it is critical to understand the scene in detail, such as in the case of autonomous vehicles, where the vehicle needs to identify and avoid various objects, and in medical imaging, where the identification and location of abnormalities are critical. Classification and localization together provide the foundation for other higher level computer vision functionalities such as object detection that is an integration of these two functionalities to provide a complete analysis of scenes.

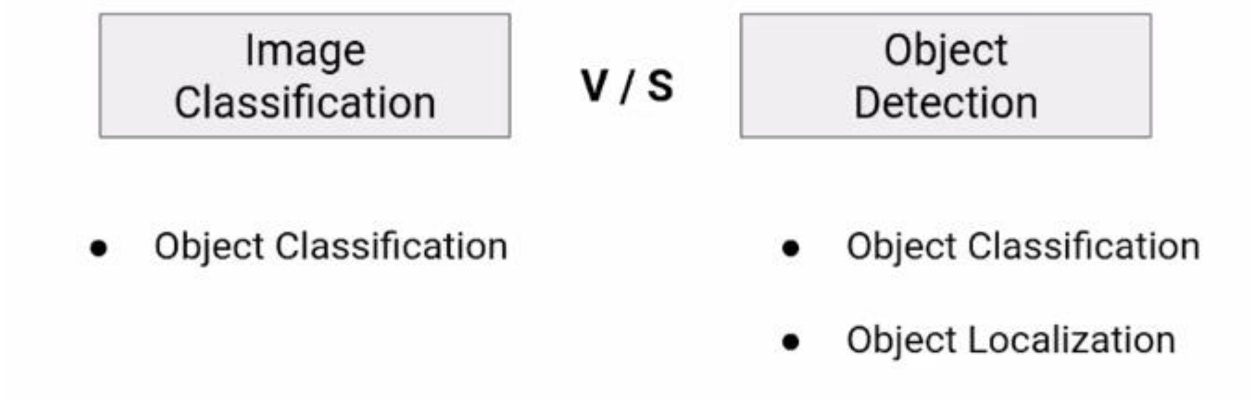


Figure 14 Image Classification and Object Detection

2.3.2 Object Detection Classification

State-of-the-art methods for object detection can be categorized into two main types: The two broad categories of estimators are the one-stage methods and the two-stage methods.

One-stage methods: In one-stage methods in object detection, the main focus is to optimize the inference time and achieve it by eliminating the need to perform object localization and classification in two separate passes in the neural network. Some of the widely known methods include YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector) and RetinaNet. These models are intended to quickly analyze image data and find objects, so they are suitable for applications where real-time operation is required, such as autonomous vehicles, security

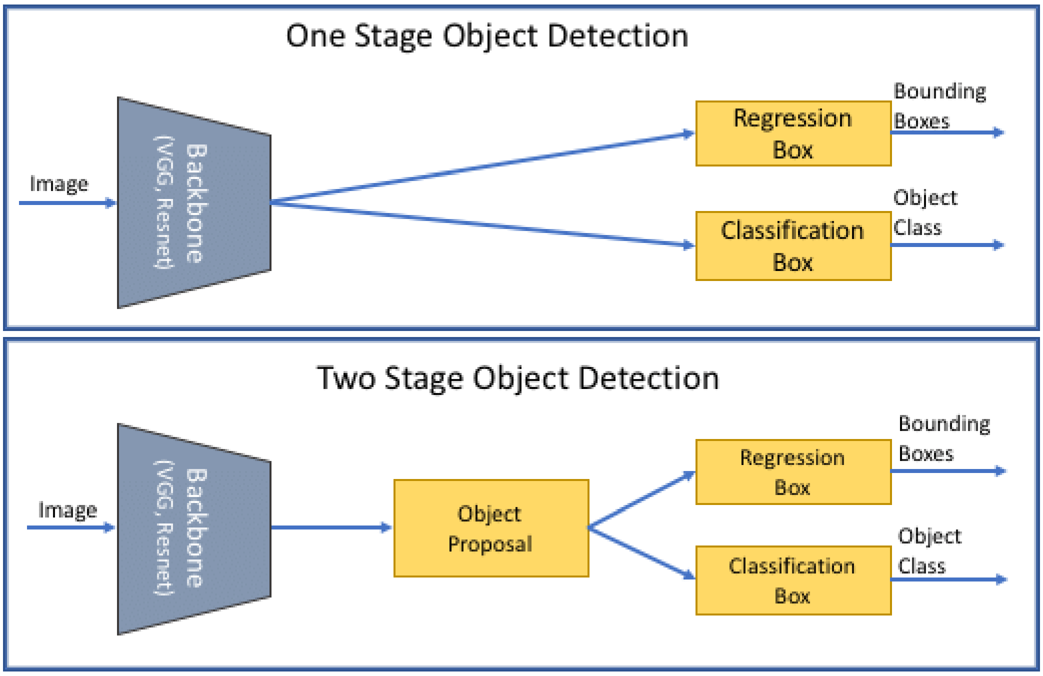


Figure 15 One-Stage and Two-Stage Object Detection [76]

systems, and augmented reality. For example, in YOLO, the image is divided into a grid and the coordinates of the bounding boxes and class probabilities are predicted directly from the cell in the grid, which makes the detection very fast [29]. SSD also partitions the image into grids but, it utilizes multiple feature maps of different scales to effectively detect objects of different sizes with high precision. To overcome the class imbalance issue, RetinaNet proposes the Focal Loss which enhances the detection of small and challenging objects. One-stage methods are characterized by the unified approach toward the detection of objects, which differs from two-stage methods that first create region proposals and then classify them as objects.

It can be several times faster, although this is often done at the expense of a slight decrease in accuracy. However, they cannot be dismissed due to their fast efficiency particularly where timely decision is important.

Two-stage methods: One of the methods used in object detection is two-stage methods because they focus on detection accuracy and use a more elaborate process than one-stage methods. Some of the famous methods that have employed these techniques include Faster R-CNN, Mask R-CNN, and Cascade R-CNN [30]. In the first stage, these models produce region proposals, which are potential Bounding Boxes containing an object. In the second stage, these proposals are filtered and regrouped, and the location of the bounding boxes is modified for better precision and accuracy [31].

These two-stage methods are more useful in cases where the detection must be accurate, for example, in medical diagnosis, self-driving cars and precise image analysis. Although they are slightly slower than the one-stage methods because of their more complex processing pipeline, the enhanced accuracy they deliver in most cases makes it worth the added computational effort.

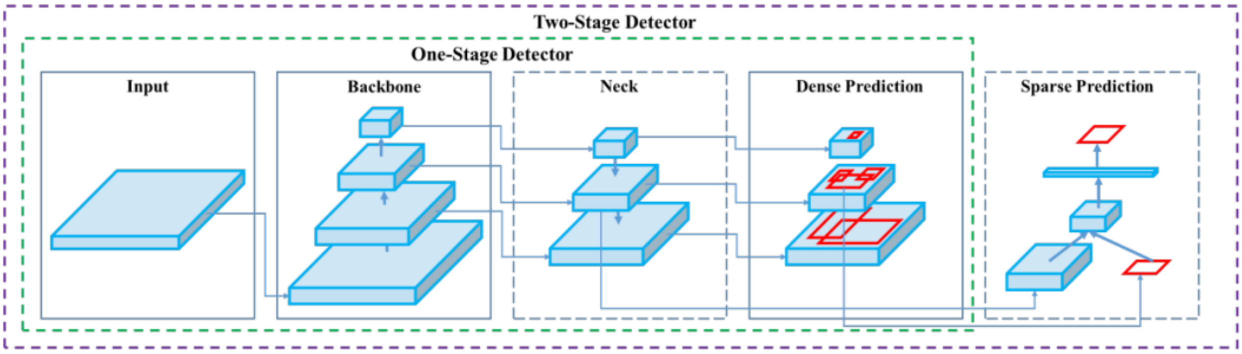


Figure 16 Two-Stage Detector [32]

2.4 Literature Review on PCB Defect Detection Using CNN

Printed Circuit Boards (PCBs) are one of the most critical components of the performance and durability of electronics products and, therefore, require thorough inspection for defects during production. This literature review focuses on the development of the method for PCB defects detection, starting from the traditional visual inspection, image processing, and machine learning, and finally the breakthrough introduced by CNN.

2.4.1 Manual Defect Detection Methods

First, the PCB defect inspection was done by hand, where operators used magnifying glasses or a microscope to look for defects such as missing components, misaligned components, and soldering problems. The first benefit of this method is that it is relatively cheap and can be done with simple apparatus. Even more experienced operators can utilize their knowledge to identify thin-sliced defects that automatic systems can fail to detect.

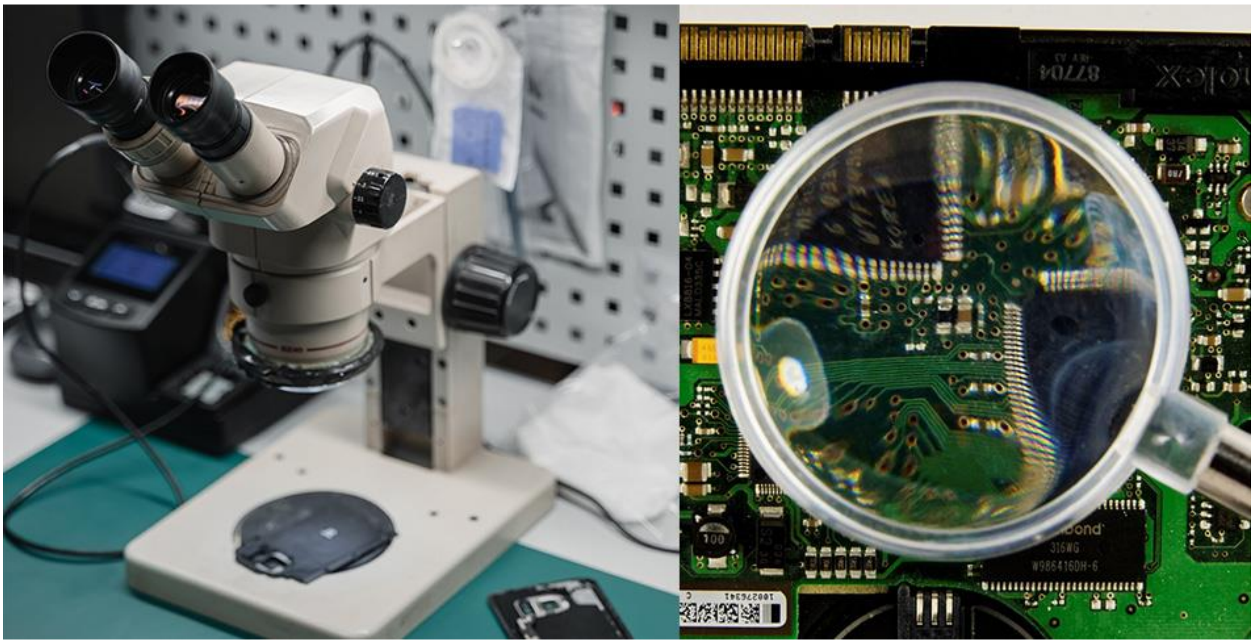


Figure 17 Manual Defect Detection Methods [33]

However, manual inspection is slow and requires much effort and time, and the results may differ depending on the operator's experience and focus. This is exacerbated by fatigue, which leads to reduced reliability, making manual inspection impractical for high-volume production. These limitations have led to the shift of the industry towards the use of automated solutions.

2.4.2 Image Processing-Based Defect Detection Methods

The advent of image processing techniques marked a significant improvement in PCB defect detection. These methods use computer vision and image processing algorithms to automate the inspection process, thus reducing human error and improving efficiency.

Reference Comparison Methods:

This approach involves comparing an image of the PCB under inspection with a reference image. Techniques like image subtraction highlight differences between the two images to identify defects [34]. While straightforward and effective for detecting gross defects, this method requires a high-quality reference image and can be sensitive to variations in lighting and positioning.

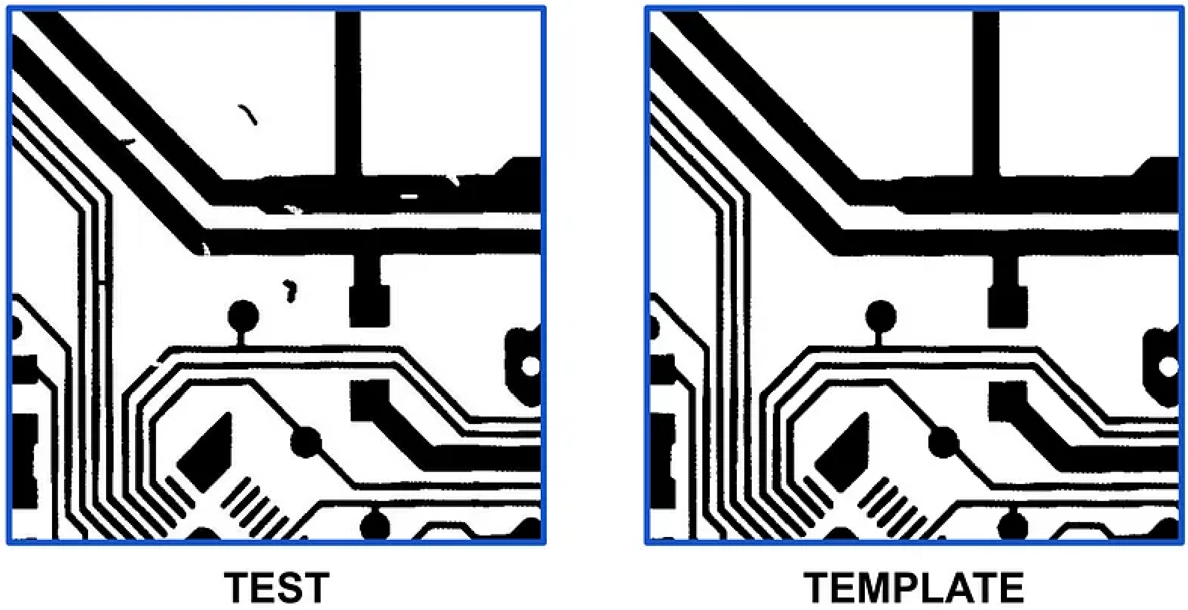


Figure 18 Reference Comparison Method [77]

Non-Reference Inspection Methods:

Unlike the previous methods, these approaches do not require a reference image. However, algorithms work on the image of the PCB and compare the image with the rules and patterns set for it, and the techniques such as edge detection, template matching, and morphological operations are used in the process [35]. While they are more flexible and less sensitive to changes in the PCB designs, these methods are normally very sensitive to parameters and may be very complex to implement.

Hybrid Methods:

As a result, hybrid methods have higher detection accuracy and are more resistant to negative influences compared to reference and non-reference methods. These methods are more useful when dealing with intricate PCBs and where the manufacturing conditions are not standard.

Automated Optical Inspection (AOI) systems commonly employ these images processing techniques, operating continuously to provide quantitative assessments of PCB quality, making them ideal for high-speed production lines [36].

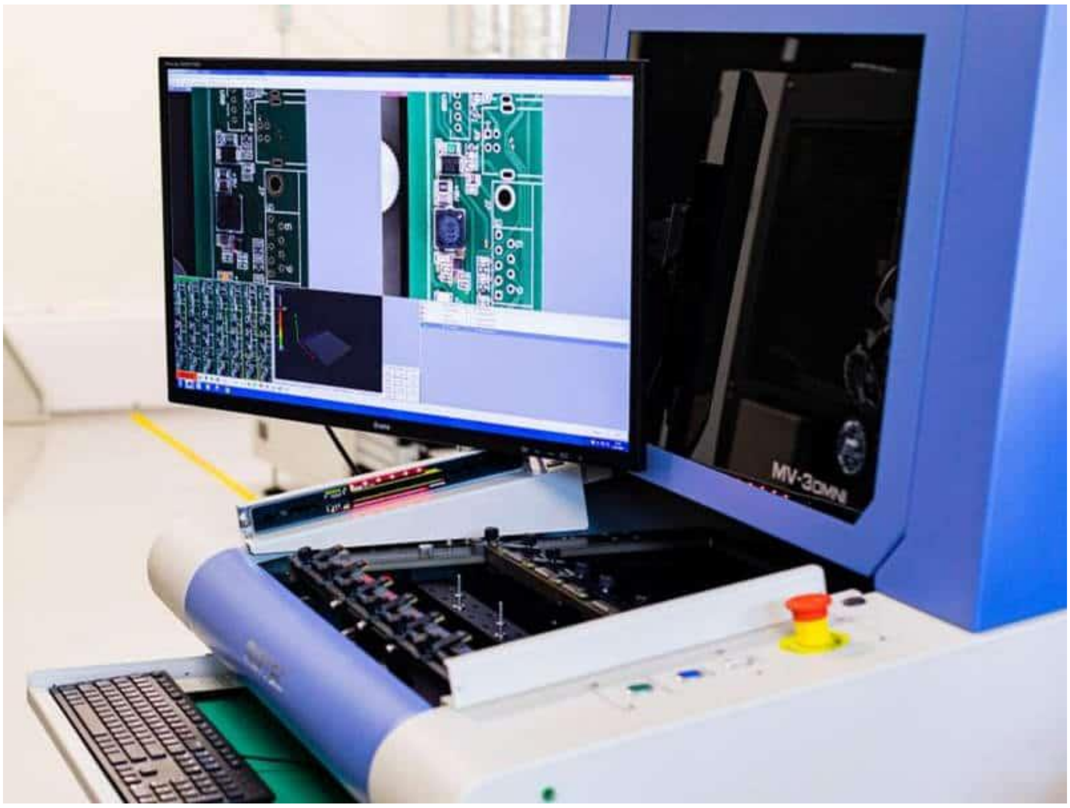


Figure 19 Automated Optical Inspection (AOI) [37]

2.5 Review on PCB Defect Detection

Printed Circuit Boards (PCBs) are critical components in modern electronics, and ensuring their defect-free production is crucial for the reliability of electronic devices. Recent advancements in deep learning and computer vision have significantly improved the methods for PCB defect detection. This literature review consolidates findings from several key papers in this domain, highlighting the progress and methodologies employed.

2.5.1 Comparative Studies and Performance Analysis

The studies conducted in the past few years focus on the performance comparison of different techniques in the identification of PCB defects. For instance, Gaidhane et al. (2018) suggested a similarity measurement method to detect local defects, while Tsai et al. (2019) utilized global Fourier image reconstruction for transformation of translation and illumination [38], [39]. In another approach, Abdel-Aziz Hassanin et al. presented a real-time method using SURF features and morphological operations [40]. Although these methods are quite useful and efficient, they have very high application conditions, and are not very robust and real-time.

For small defect detection, two-stage object detection networks such as Faster RCNN have been suggested, with Hu and Wang (2020) and Yuan et al. (2024) offering their frameworks [3], [41]. However, these methods have some problems, such as complex training procedures and low detection rates. Some of the limitations of two-stage object detection algorithms are as follows: Kang et al. (2021) and Bhattacharya et al. (2022) have overcome these problems by using one-stage object detection algorithms by reducing the framework of the network and increasing the speed of inference, but in high-precision small target detection, they may have problems [42], [43].

The changes in the methods of detection of PCB defects reflect the progress in technology and the need for better quality of electronic production. Starting from the traditional direct visual assessment up to the modern CNN-based approaches, all the methods have been enhancing the accuracy, speed, and robustness of the defect identification [44]. CNNs have become the most promising solution due to their capability of learning features and addressing sophisticated vision problems. Future studies will probably be devoted to improving these methods and using them in combination with other innovative technologies as well as to exploring new issues in the manufacturing context.

2.5.2 MobileNet-Yolo-Fast and YOLOv8 in PCB Defect Detection

In Liu and Wen (2021), a defect detection framework using MobileNet-Yolo-Fast was proposed, which integrates the MobileNet for its lightweight structure and YOLO for real-time defect detection in PCBs [45]. This approach greatly decreases the computational burden while preserving the high detection accuracy, which is particularly valuable in industrial applications where speed and resource consumption are important factors.

Xiong (2023) put forward an improved framework by integrating YOLOv8 for detecting bare PCB defects. The YOLOv8 model provides better accuracy and response time compared to its predecessors, and it uses more features such as a deeper network and optimized loss function [46]. This system was able to detect several types of defects such as scratches and missing components which was a better performance than the previous YOLO systems.

2.5.3 Dataset and Classification Techniques

Huang and Wei (2019) have proposed a detailed PCB dataset for the identification of defects and its categorization, which is crucial for the evaluation of the detection models [47]. It consists of different types of defects including open circuit, short circuit, missing hole etc., and is used for the evaluation of different detection techniques.

Adibhatla et al. (2020, 2021) have proposed the use of convolutional neural networks based on YOLO for the detection of PCB defects. They dedicated their research on improving the YOLO design with the aim of improving the localization and classification of defects [48], [49]. According to them, there was a considerable enhancement in the detection rate and a decrease in false positives making YOLO a reasonable choice for PCB inspection activities.

2.5.4 Enhanced YOLO Models

Du et al. (2023) proposed YOLO-MBBi which is an improved model based on YOLOv5 for detecting surface defects of PCBs. The YOLO-MBBi model incorporates multi-scale feature extraction and a balanced beam intersection module that makes it capable of recognizing small and intricate defects [6]. In their experiments, they proved that the proposed YOLO-MBBi is more precise and has a higher recall rate than the traditional YOLOv5.

Also, Tang et al. (2023) proposed the PCB-YOLO model, which is a better detection algorithm than YOLOv5 [7]. This model introduces more contextual information and tuned hyperparameters to enhance the detection rate of defects, especially in conditions where they are small or obscured.

2.5.5 Transformer and Lightweight Models

Chen et al. (2022) developed a Transformer-YOLO model that leverages the transformer network and YOLO to identify defects [50]. The transformer part increases the model's capacity to capture long-range dependencies and contextual features, which helps to achieve higher accuracy in identifying minor flaws on the PCBs.

As part of the effort to create lightweight models, Zhou et al. (2024) introduced a new detection algorithm called MSD-YOLO [51]. The MSD-YOLO algorithm is optimized to be used in devices with limited computational power, which makes it effective for real-time use in production. The model is able to reach a high detection accuracy without a large computational cost.

2.5.6 Contextual and Attention Mechanisms

Xia et al. (2023) introduced a global contextual attention mechanism augmented with ConvMixer prediction heads in their YOLO-based model [52]. This approach enhances the model's focus on relevant regions of the PCB, improving defect detection rates by effectively distinguishing between defects and background noise.

Jiang et al. (2023) presented the DCR-YOLO model, which employs a dual-channel residual network to enhance feature extraction capabilities [53]. This model is particularly effective in detecting surface defects, achieving high precision and robustness against variations in lighting and PCB surface conditions.

2.5.7 Classical Approaches and Early Research

Leta et al. (2008) discussed a computer vision system for PCB inspection using classical image processing techniques [54]. Although not as advanced as contemporary deep learning methods, their work laid the foundation for automated PCB inspection by highlighting the importance of high-resolution imaging and precise defect localization.

Xin et al. (2021) focused on an improved YOLOv4 algorithm for detecting electronic component defects on PCBs [4]. Their enhancements to YOLOv4 included better anchor box selection and modified loss functions, which resulted in improved detection accuracy for component-level defects.

2.6 Survey and Future Directions

Ling and Isa (2023) provided a comprehensive survey of PCB defect detection methods, encompassing image processing, machine learning, and deep learning approaches [55]. Their review highlighted the rapid evolution of defect detection techniques and the increasing adoption of deep learning models, which offer superior accuracy and automation capabilities .

Chen and Xie (2023) explored a machine learning approach for automated detection of critical PCB flaws using optical sensing systems [56]. Their study emphasized the integration of machine learning with optical inspection technologies, achieving high detection accuracy and real-time performance.

2.6.1 Drawbacks of PKU-Market-PCB Dataset

The following are the limitations that are associated with the dataset which makes it unsuitable for use in real-world PCB defect detection: Firstly, it is limited to only 10 PCB boards, which may not be representative of the full range of PCBs that are manufactured in real life [57]. Second, the defects in this dataset are introduced after manufacturing through applications such as MS Paint, and hence, they do not closely resemble real-world manufacturing imperfections, which makes this dataset less suitable for training models for realistic industrial applications. Furthermore, each PCB board is labeled with only one fault class while in real life, a PCB board might have multiple faults at a time which makes the dataset less realistic to real life manufacturing scenarios where boards are likely to have multiple faults. These limitations are compounded in aggregate to suggest that the dataset is not well-suited for generalization to various and complex situations that one might encounter in real-world PCB inspection tasks.

2.6.2 Drawbacks of Deep-PCB Dataset

The dataset used for PCB defect detection has several limitations which should be considered in terms of the applicability of the work carried out in the present study to real-world situations [57]. Firstly, it uses cropped-up images of a single PCB, which might not capture the entire scope and the range of layouts and defects that a real-world PCB might have. This limited perspective could lead to the model not generalizing so well across different PCB designs and fault patterns. Second, the dataset does not include the full dimensional depth of PCB design schematics, which may limit the model’s ability to learn from a variety of layouts that may be observed in real-world applications. Also, the datasets limit images to only black and white, thus reducing the complexity of the defect identification process but at the same time ignoring the color variations that exist in real world images of PCBs. This simplification might reduce the model’s generalizability since, in the real world, PCBs contain components and markings of different colors, which provide essential information for correct defect identification. Altogether, these limitations highlight the dataset’s shortcomings in training models for the real-world and its variations characteristic of most PCB inspection scenarios.

Chapter 3

Methodology

Printed Circuit Boards or PCBs can be found in almost all electronic devices and the process of making them involves a lot of detailing to guarantee that the end product will be strong and reliable. Chemical etching can be said to be one of the most important steps that help define specific circuit patterns because it provides the means of selectively removing the copper. This chapter deals with the process of developing a PCB defect identification dataset using chemical etching and other approaches involved in this process are explained in this chapter.

Chemical etching is the most critical process in creating electrical circuits in PCBs due to its sensitivity; it is vital to achieve maximum precision. Photolithography is used to draw an image of what needs to remain and what needs to be etched away from the copper while keeping the electronics. Though this step has to be very proper and sophisticated, it is a potential source of many errors that may negatively affect the functionality of the PCB. In order to make a good collection for defect detection, one has to know about the differences in dry and wet etching methods. Dry etching is more selective and uses plasma or lasers for etching but it is expensive and cannot be utilized in mass production. Wet etching is more preferred especially the alkaline method because it is efficient and cheaper to pursue. To get rid of unwanted copper, chemical solutions are treated under considerable pressure within a reactor. This should be well done to avoid any harm as well as to also ensure that the quality is achieved. PCB flaw identification dataset creation is a multi-step process commencing from etching up to the review phase. After etching, the circuitry is washed to ensure that it is free from etchant and the ink which was used in the etching process. This ensures that the circuitry is closed and ready for inspection or other subsequent operations that it may be required to undertake. Subsequently, PCB is coated with an epoxy resin to protect it from the climate and to avoid short-circuiting. This makes it have a longer durability and be able to perform the intended function better than other materials that are not subjected to such conditions.

This chapter will describe the means used to obtain a dataset that would be capable of identifying the errors that occurred during chemical etching. It will also include the types of faults most used, how to record these imperfections and how to label it, and how to make this information as complete as the real world. Therefore, the better the datasets are informed about the chemical etching process and the mistakes that can occur during it, the more suitable are the datasets for enhancing the models of flaw detection. It will also assist in enhancing the quality and reliability of PCBs within the electronics industry in the long-run.

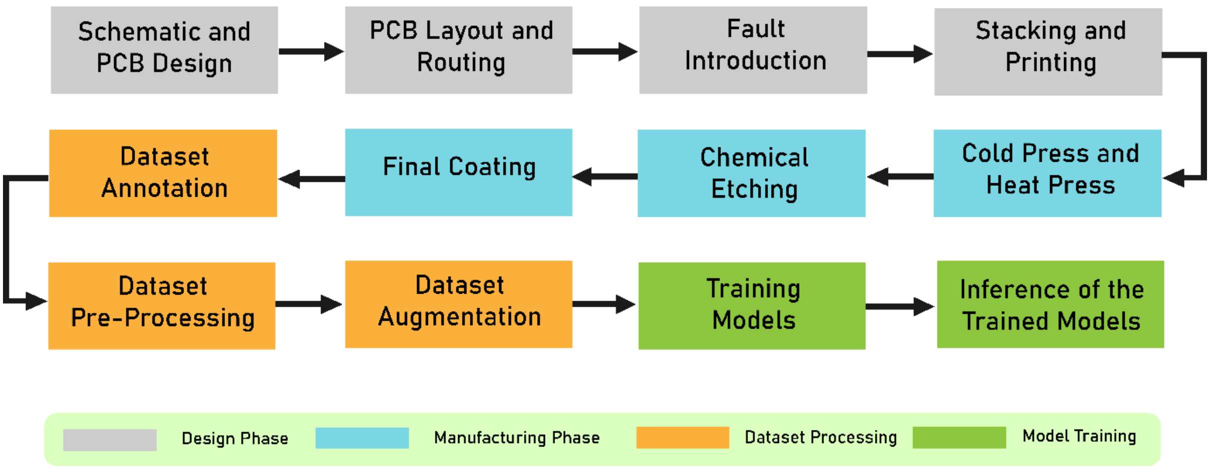


Figure 20 Research Methodology

3.1 PCB Manufacturing Using Chemical Etching

3.1.1 Schematic and PCB design

The first step in the actual design of the PCB is to create the rough layouts with the help of PCB designing softwares such as KiCAD and EasyEDA. This is a crucial stage where design symbols, simulation models, PCB footprints, and 3D step models within the CAD library are developed. Each part in the library has been designed in a manner that ensures that it closely resembles the real part and is compatible with the rest of the circuit design. This is the step that provides the basis for the entire planning of PCBs and the manufacturing processes.

However, circuit modelling is also crucial in ensuring that the blueprint design is correct or not. Before moving on to the PCB planning step, the simulation of the circuits using tools like SPICE comes after the planning. This advance confirmation is very useful before the PCB is made because it is easier and more efficient to identify and correct any design problems that may exist as early as this stage.

3.1.2 PCB Layout and Routing

When the schematic design has been done, and reviewed, the focus shifts to the PCB planning stage. The layout artist imports the connection data from the schematic into the PCB layout software. This information is converted into nets, which depict the connectivity of the electrical links between pins on a component. While installing components, much attention is paid to the positioning of the PCB footprints in the most optimal way according to the shape of the board. This is done to obtain the most efficient routing paths with the least electrical interference, thermal issues, and signal issues. Layout designer utilizes CAD tools that have design rule check (DRC) to ensure that part is placed within specified tolerance and route is according to the laid down rules.

In the wiring process, PCB lines and planes are employed in connecting component pins. For the fast realization of this process without a negative impact on the quality of the design, high-level CAD tools are employed, including automatic routing techniques. A lot of care needs to be taken while laying signal lines so that the signals are not distorted and there are no couplings or interference between the parts of the circuit.

3.1.3 Introduction of Faults and PCB Layout

Once the planning and wiring of the PCBs are completed, then the boards are prepared for the introduction of faults and testing. To achieve this, the PCB plans need to be converted to SVG files for Adobe Illustrator to incorporate six pre-coded fault types. These defects; such as missing pads, open circuits, shorts, spurs, fake copper, and mouse bites are intentionally designed and purposely placed on the PCB designs.

The next step is to print the PCBs so that the ink can be transferred onto the surface of the board. To create a single PCB, several PCB plans are overlaid to create an A4 layout plan. Laser printers then print this layout on photo paper. The layout of the photos on the photo paper should be such that it can accommodate all the photos to be developed. Prior to the ink transfer, the PCBs are cleaned with alcohol pads really well, removing any dirt or any other material that could interfere with the binding of the ink.

3.1.4 Cold and Heat Press Preparation

To ensure that the ink adheres well on the board surface, the PCBs undergo a process known as Cold Press and Heat Press The boards are dipped in a solution of 60% isopropyl alcohol and 40% acetone so that printed A4 papers can be placed on them during Cold Press. The setup is then exposed to a mounting mode laminating roller which ensures that the ink bonds well to the surface of the PCB.

Cold Press is followed by boards being left to dry for 30 minutes to ensure that the ink properly adheres to the PCB surface. Subsequently, the laminating machine is utilised during the Heat

Press phase where the heat settings are activated. After the printing, the PCBs are heated and pressed again and again, 25 times to be exact, in order for the ink to become more sticky so that it will adhere well on the board surface. This ensures that the coloured traces are well secured and can endure shocks from the environment as well.

3.1.5 Chemical Etching and Final Coating

Chemical etching and the applying of a protected covering are the final stages in preparing a PCB. It is now possible to etch the PCB boards using a ferric chloride solution that is prepared using a certain quantity of water. This is the case where the intentional faults are highlighted using ink. This particular chemical etching process allows the removal of copper only in the areas that are not masked by ink. This makes the circuit lines clear in accordance with the design plan in question.

In addition to etching, the PCBs are washed very well to remove any ink and etchant that may still be on the surface of the board. After this, an epoxy resin covering is applied on the PCBs for protection from other factors, moisture, and formation of short circuits. Epoxy glue also increases the thermal conductivity of the PCBs and this is very crucial in the dissipation of heat that electronic parts produce when they are functioning.

3.1.6 Factors to Consider for High-Quality Etching

Thus, the printing process must be closely monitored, and the quality of the circuit boards ensured by improving the printing process. It is also important to understand that the etching process has its effectiveness and dependability highly dependent on a number of factors.

First of all, the concentration of chemicals in the etchant solution should be optimal to ensure proper etching of the material. The right proportion helps the etch rates to be stable and uniform so that the copper traces of the PCB do not become too thin or too thick. To maintain such ideal conditions throughout production, it is required that the etchant solution be monitored frequently and replenished as needed. Second, it is very important to maintain a constant temperature during etching as it has a significant impact on the process. Elevated temperatures can work faster to etch the material, at the same time, the etchant is consumed at a faster pace. The temperature is thus required to be regulated to achieve a balance between effective etching and economical use of chemicals. This helps to make the process stable and cost effective. There is also the manner in which the etchant solution and boards are transported or handled from one place to another. This makes certain that fresh etchant reaches all the copper's surfaces so that the etching process is done effectively. This prevents the etchant from draining

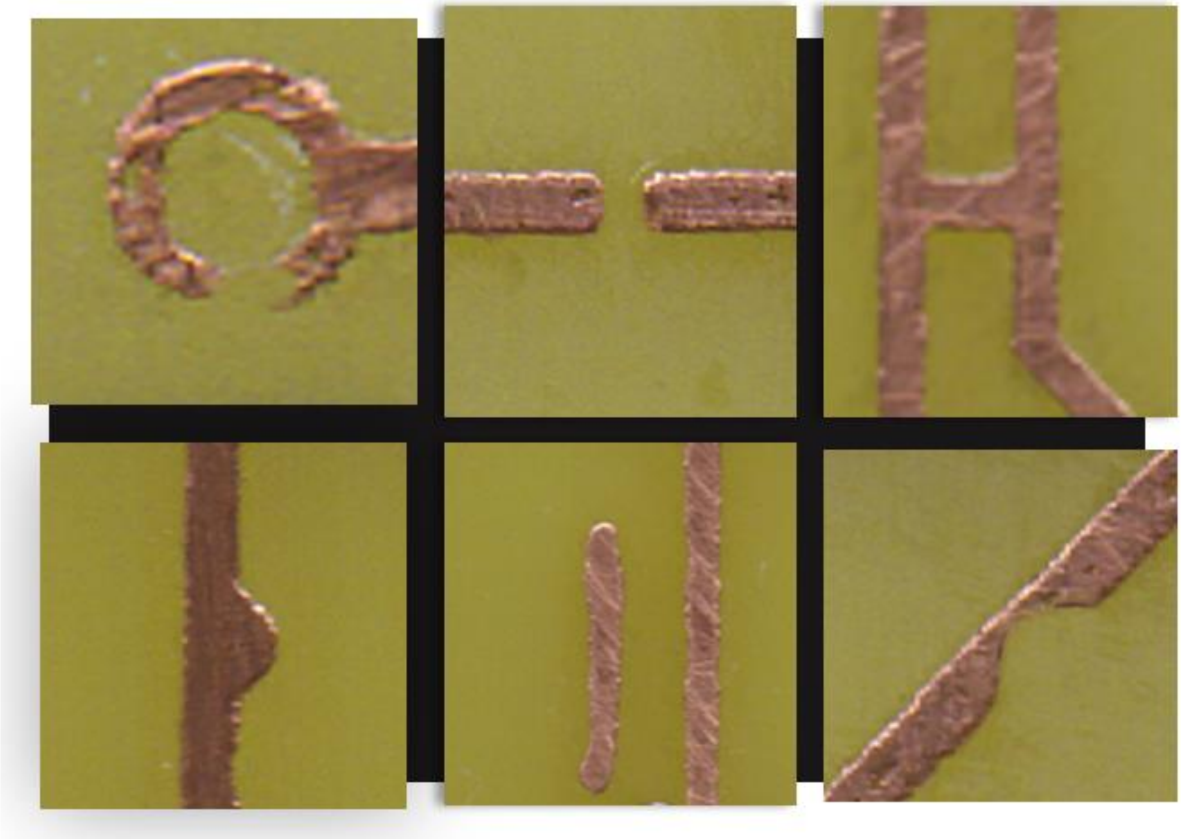


Figure 21 Defects Manufactured in the Dataset

at some parts and this leads to bad etching of the PCB lines. For the etching conditions to be the same on the boards being worked on, there is need to have good motion systems.

The quality of the end circuitry is an indication of the etch rate, which is the rate at which copper is removed from the PCB. Making the etching process faster might increase the output, but it may also lead to more peeling off of the resist material or improper removal of pad areas. The etch rate is another crucial factor that determines the extent to which copper is removed from the board and, therefore, the functionality of the resulting PCBs. It is also very important that the prevent material used to protect the copper lines is of the highest quality. It is also important for the resist to be uniform and smooth, and any imperfections such as pinholes or thin areas can compromise the ability of the resist to shield the copper from being etched. Ensuring the quality of the resist application and checking methods reduces the probability of etching copper unintentionally and enhances the general quality of the PCB production line.

Moreover, it is extremely important that all the PCBs bear the same marking so that there could be a regular and steady production of circuit boards. Variations in the etching process can result in imperfections or roughness in the lines of the PCB and this has an impact on the reliability of the electronic circuit. Concentration, temperature, stirring, and etch rate are among the vital etching factors that should be monitored closely to ensure that the production of PCBs continues to meet the industry standards.

3.2 Data Processing Method

3.2.1 Dataset Annotation

The dataset used in these experiments is a set of 230 images where every picture is carefully marked with bounding boxes of six different classes of objects. In general, it is possible to estimate that there are about 7. 5 annotations, thus helping to create a more diverse and inclusive set of resources. In terms of image sizes, they are not standard and can be as small as 2. 57 megapixels to 31. 26 megapixels, with an average size of 6. 61 megapixels. The median image resolution is 2663 x 2587 pixels, and it ranges from (800 x 600) to (6000 x 4000). This variability makes sure that the dataset can contain a lot of real-world conditions, thus improving the detection models.

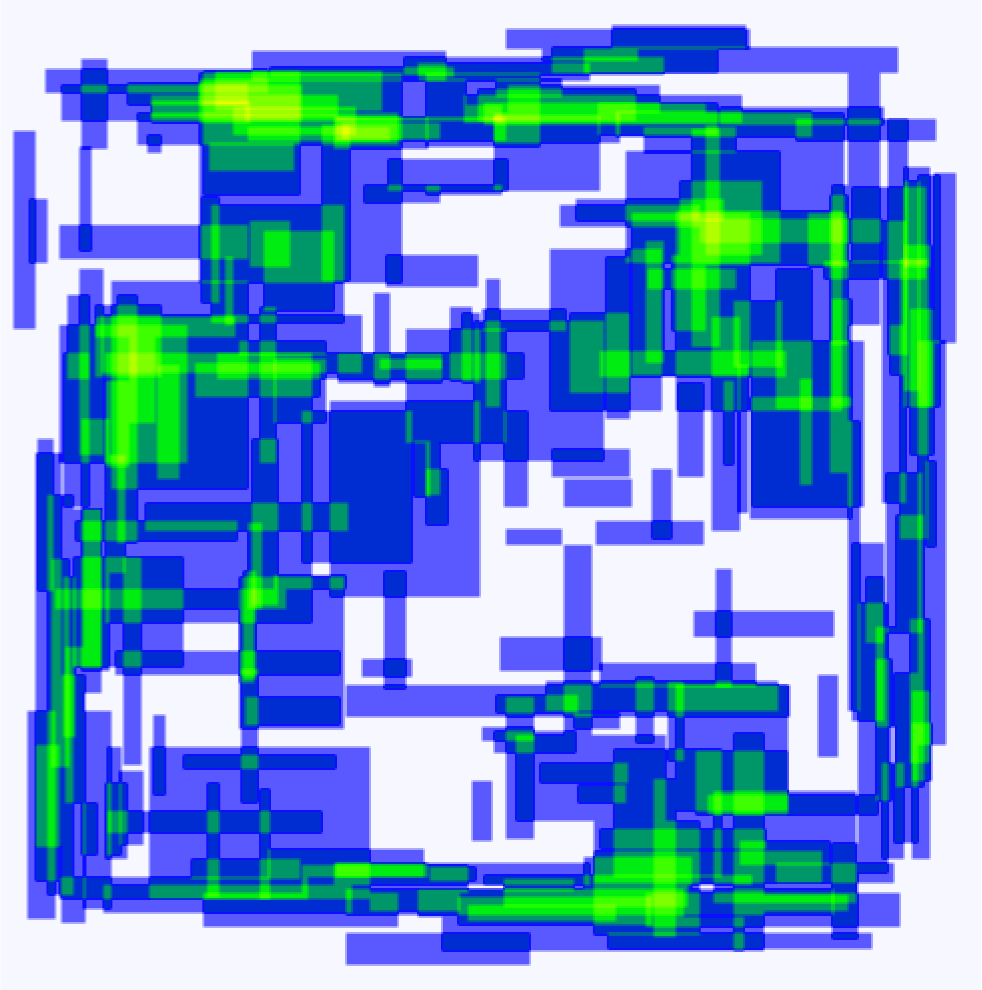


Figure 22 Annotation Heatmap

3.2.2 Annotation Heatmap

The annotation grid of the PCB dataset shows the distribution of annotations and the density of the same in the dataset. The heatmap pinpoints areas of different density of annotations and the green coloration means high density. These high-density areas usually relate to areas of high component or feature density on the PCB which is desirable for specific detection tasks.

On the other hand, regions coloured in blue are those that are less annotated, which can mean that those parts of the PCB are either unimportant for the detection tasks or contain fewer features. The identification of such low-density regions is essential to fix the problem of imbalanced data distribution and to design effective solutions like data enrichment or balanced sampling in order to improve the model training. The given labels distribution helps to understand the general distribution of the dataset and avoid over-emphasis on specific PCB areas during the training process, thus improving the model's recognition capabilities and preventing overfitting.

Table 1 Total Annotations Per Class

Total Annotations per Class	
Mouse-Bite	269
Spurious-Copper	247
Spur	257
Open-Circuit	261
Short-Circuit	242
Missing-Pad	244

3.2.3 Data Pre-Processing

Data pre-processing is a crucial step before feeding the data into the model to improve the model's performance and cut short the training time. To accomplish this, all images in the PCB dataset were pre-processed through a sequence of transformations.

Resizing: All images were resized to the dimension of 1800 by 1800 pixels to keep all datasets uniform in size. To keep the images as close to the original size as possible, bars in black color were added where necessary and this was done to ensure that the visual content is not distorted in any way.

Augmentation: Offline data augmentation was done as a preprocessing step to enhance the model's ability to generalize and reduce training time. This is because offline augmentation leads to consistent image transformations that enrich the dataset while not hindering the training process. The augmentation techniques applied are as follows:The augmentation techniques applied are as follows:

- Flipping: Mirroring was performed both along the horizontal and vertical axes to increase the variability of the images while maintaining the consistency of the annotations.
- Rotation: Images were also mirrored horizontally, and vertically flipped; images were also rotated 90 degrees clockwise, and 90 degrees anticlockwise.
- Random Rotation: To add more variability to the images, the images were rotated randomly by -15 to + 15 degrees.
- Shearing: Images were rotated up to ±10 degrees along the horizontal as well as the vertical axis.
- Brightness Adjustment: Lighting conditions were changed to -15% and + 15% to test the ability of the system to adapt to different brightness levels.
- Exposure Adjustment: Gamma exposure ranged from -10% to +10%.

- Blurring: To mimic the effects of blurring due to out of focus, up to 2 pixels of Gaussian blur was applied.
- Noise Addition: It was found that salt and pepper noise up to 0. 22% of the pixels are used to mimic the different levels of image noise.

To connect these augmentations, randomization was used, and this helped in creating diverse training images. This process created multiple improved copies of every initial image, which greatly increased the size of the dataset after filtering out similar images.

3.3 Model Architecture

3.3.1 YOLO Architecture

YOLO's design is based on dividing input images into a S x S grid. Each grid cell is in charge of finding items whose centres are in its own cell. This grid-based method not only speeds up processing by letting multiple areas of space be evaluated at the same time, but it also makes it easier to understand the context, which is important for finding objects correctly.

YOLO suggests several bounding boxes for each grid cell and gives confidence scores based on how likely it is that each box contains an object, which is found by multiplying the object presence probability (Pr(Object)) by the intersection over union (IoU) with ground truth boxes [58]. This choice in design makes it possible for YOLO to work well in situations where there are multiple objects in a cell, improving detecting performance across a range of spatial sizes and object densities. YOLO also predicts conditional class probabilities in each grid cell. This makes the classification job easier by separating it from object location. This makes the model more flexible in recognising different types of objects.

The backbone of the design uses CSPDarknet, which has been customised with certain depth and width factors and includes spatial pyramid pooling (SPP) layers with different kernel sizes [59]. These features allow YOLO to identify hierarchical features at different scales, which is necessary for finding things in pictures that are different sizes and levels of complexity. In addition, smart choices in the normalisation and activation functions of the backbone keep the training dynamics stable and improve the ability to represent features.

YOLO has a YOLOXPAFPN neck module that gradually joins feature maps from different backbone levels to improve the extracted features and make recognition more accurate. Before going into the detection head, this integration improves the spatial precision and semantic complexity. This makes sure that the whole object is understood and its location is accurate. The detecting head is made up of stacked convolutional layers that are best at guessing bounding boxes and the confidence scores that go with them on a number of different scales. Loss functions designed for objectness scoring, bounding box localization precision (IoU-

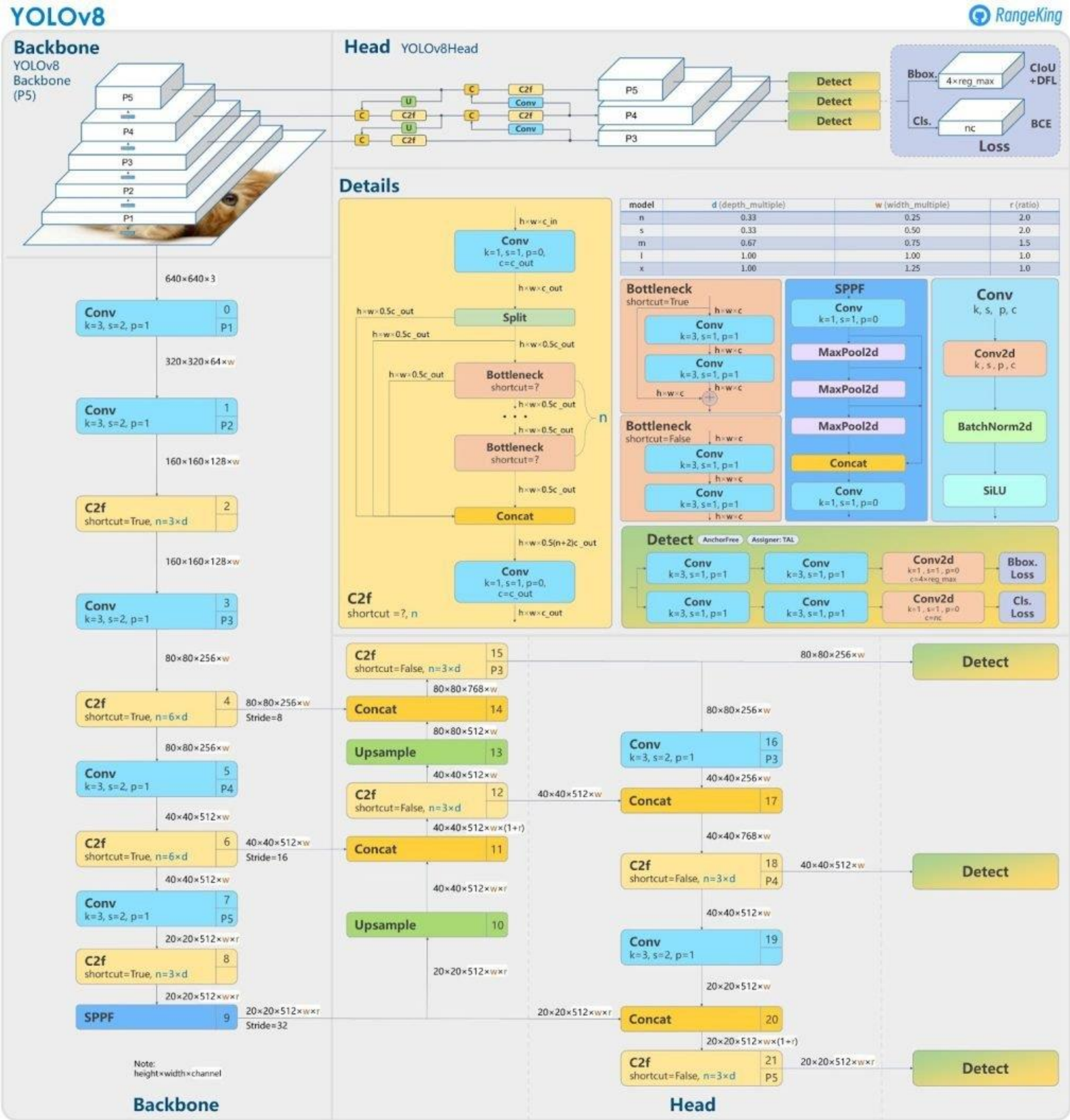


Figure 23 YOLOv8 Architecture [78]

based), and classification accuracy improve predictions even more during both the training and inference stages. This makes the system work well with a wide range of datasets and situations.

During training, YOLO uses a loss function with many parts to even out mistakes in classification, localization, and estimating confidence. During deployment, YOLO uses non-maximum suppression (NMS) to get rid of unnecessary detections [60]. This makes sure that each item is accurately named and located.

YOLO uses a stochastic gradient descent (SGD) planner with momentum and weight decay, along with a dynamic learning rate schedule that includes warm-up times and cosine annealing, to improve performance and make it easier to scale. This plan makes it easier for the model to get closer and closer to the best answers over the course of training epochs. Extra features like mode switching and exponential moving average (EMA) help keep the model running smoothly and make it better at generalization [61].

3.3.2 RetinaNet

ResNet uses a number of convolutional layers to record fine features and spatial hierarchies at different scales, which makes it possible to accurately locate and classify objects. A Feature Pyramid Network (FPN) is built into RetinaNet to deal with the fact that object shapes and scales can vary in pictures. This structure groups together feature maps with different levels of detail, which lets the model make a pyramid of features that fully capture meaningful information in a variety of spatial settings [62]. FPN improves RetinaNet's ability to find things at both fine and coarse levels by combining feature maps from different network lengths. This makes sure that it works well across a wide range of datasets and real-world situations. This

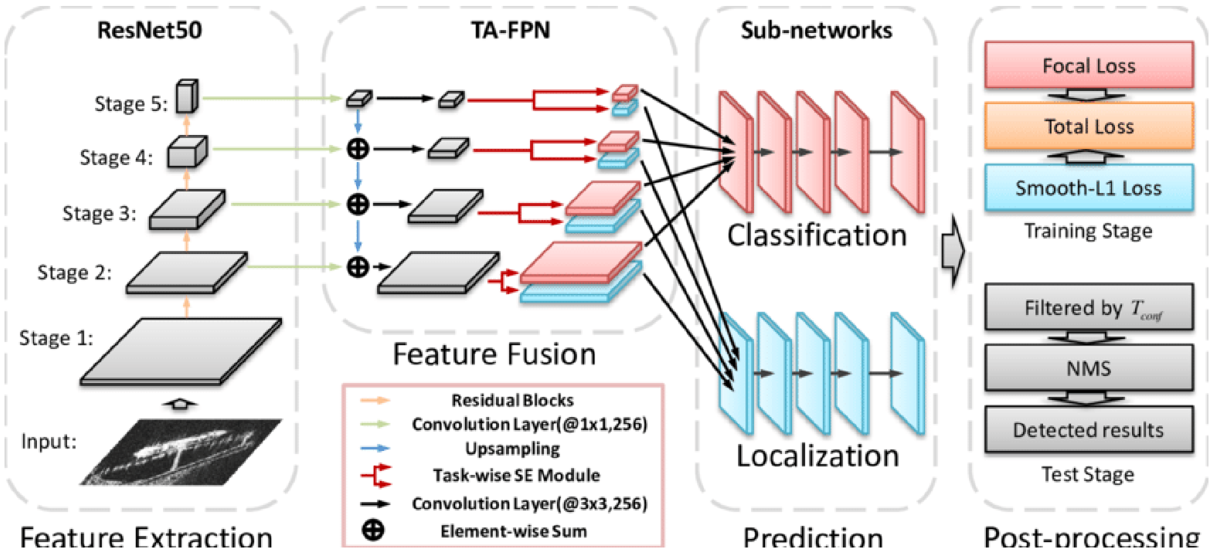


Figure 24 RetinaNet Architecture [79]

part of RetinaNet, called the GARetinaHead module, is what does most of the work when it comes to recognition. GARetinaHead predicts both the bounding box coordinates and the class odds at the same time. It was made to handle multi-class object recognition jobs. Anchors, which are fixed boxes with different sizes and aspect ratios, are used as points of reference for these predictions. There are two kinds of anchor generators used by RetinaNet. The Square Anchor Generator stays with the same aspect ratio and scale across all levels of the feature pyramid, while the Approximate Anchor Generator works with various scales and aspect ratios per octave. Loss functions in RetinaNet are very important for making the model work best during training [63]. We also use smooth L1 loss for bounding box regression with a beta value of 0.04, which reduces mistakes in guessing box coordinates and improves the accuracy of object localization. During the training process, RetinaNet uses special assignment tools to give anchor boxes ground truth labels based on which objects they intersect over union (IoU) with. A Random Sampler picks a fair group of anchors for training batches based on a positive fraction of 0.5 to keep the right mix of positive and negative samples. Additionally, centre and ignore rates of 0.2 and 0.5 make the choice of training samples better by focusing on points that are close to object centres and ignoring those that are in the background or aren't clear.

In RetinaNet, optimisation techniques are designed to keep training dynamics stable and improve convergence. To stop gradient bursts, gradient clipping with a maximum norm of 35 and a norm type of 2 is used. This keeps changes to the model parameters stable and under control during the training process.

3.3.3 ATSS Architecture

ATSS computes the IoU (Intersection over Union) between anchor boxes and ground truth objects across the dataset. Based on these IoU values, anchors are categorized into foreground and background groups, with foreground anchors being those likely to contain objects of interest. This adaptive approach ensures that training primarily focuses on anchors with higher potential for meaningful learning, enhancing the model's ability to accurately localize and classify objects.

Central to ATSS is its robust feature extraction mechanism, typically implemented using a feature pyramid network (FPN) [64]. This network structure facilitates the extraction of multi-scale features from input images, enabling the model to detect objects of various sizes and complexities. By integrating feature maps from different network depths, ATSS ensures comprehensive spatial coverage and semantic understanding, crucial for precise object detection across diverse scenarios.

The core detection mechanism in ATSS revolves around its Bounding Box Head, which refines anchor predictions by predicting bounding box coordinates and object class probabilities. The

GARetinaHead, commonly employed in ATSS, employs stacked convolutional layers to process feature maps and generate accurate predictions [65]. Anchors, generated across multiple scales and aspect ratios are encoded, which standardizes the prediction format by normalizing bounding box coordinates. This ensures consistency and accuracy in localizing objects within images. Loss functions in ATSS are carefully designed to optimize model performance during training. The focal loss mechanism, effectively handles class imbalance by prioritizing hard-to-classify examples while down-weighting easily classifiable ones. Additionally, ATSS incorporates IoU-based loss functions, to refine bounding box predictions by penalizing deviations from ground truth boxes.

During the training phase, ATSS employs specialized assignment strategies to assign ground truth labels to anchor boxes effectively. A Random Sampler selects a balanced subset of anchors for each training batch, guided by a positive fraction to maintain a suitable ratio of positive to negative samples. Moreover, training configurations such as center and ignore ratios optimize the selection of training samples by prioritizing anchors that closely align with object centroids and filtering out irrelevant or background samples. Optimization strategies in ATSS are geared towards stabilizing training dynamics and promoting convergence. Gradient clipping with a maximum norm and norm type controls ensures that updates to model parameters are controlled, preventing gradient explosions and maintaining stability throughout training.

3.3.4 NAS FPN

The NAS FPN (Neural Architecture Search Feature Pyramid Network) represents a sophisticated approach to feature extraction in object detection, combining elements from neural architecture search and the feature pyramid network (FPN) to enhance the efficiency and accuracy of models. At its core, NAS FPN integrates a ResNet-50 backbone with an FPN, utilizing batch normalization (BN) layers configured to allow gradient updates during training, crucial for optimizing normalization parameters [66]. This setup ensures that the network can effectively extract multi-scale features essential for detecting objects of varying sizes and complexities across images.

The model setup begins with data preprocessing using normalization parameters to standardize images with specific mean values and standard deviations. Images are converted from BGR to RGB format, and additional padding is applied to ensure dimensions are multiples of 64, with fixed-size padding set to maintain uniformity at (640, 640). These preprocessing steps are integral to preparing input data for consistent and effective feature extraction by the NAS FPN architecture

3.3.5 Faster R-CNN

The architecture of Faster R-CNN integrates a deep convolutional neural network (CNN), typically based on models like ResNet or VGG, for feature extraction. These networks are pretrained on large-scale datasets like ImageNet, leveraging their ability to capture hierarchical features from images effectively. The backbone network, often ResNet-50 in many implementations, extracts high-level semantic features from input images, crucial for subsequent object detection tasks [67].

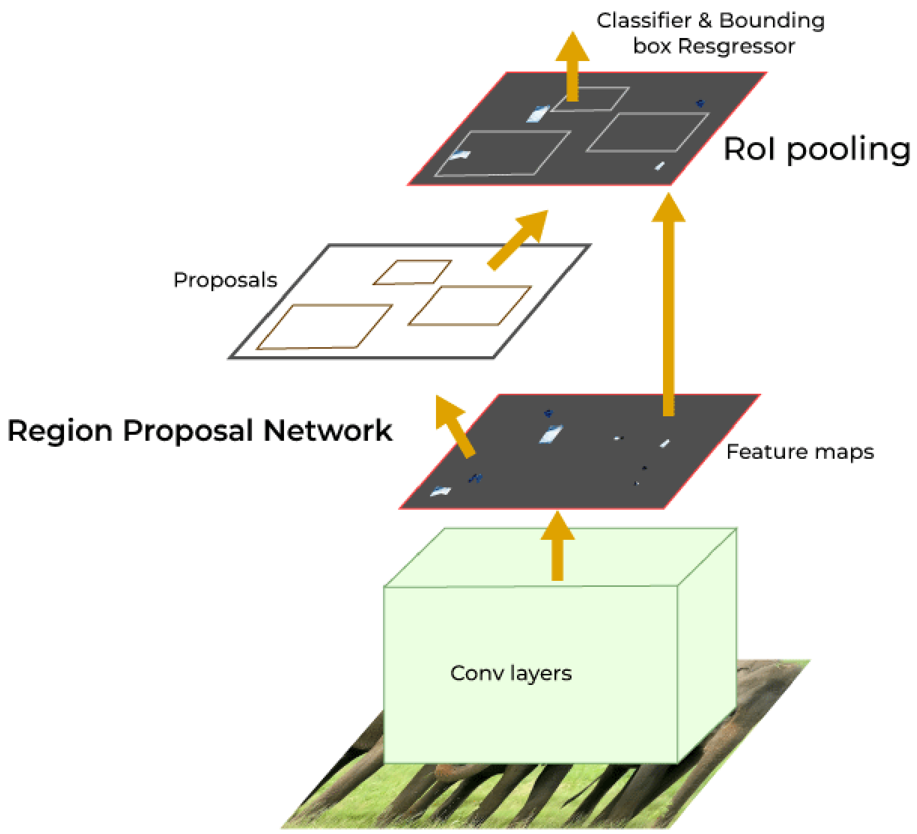


Figure 25 Faster R-CNN Architecture [80]

The RPN generates region proposals, or candidate bounding boxes, by sliding a small network over the CNN feature map. Each sliding window predicts objectness scores and refinement offsets for a set of anchor boxes of different scales and aspect ratios. These anchor boxes serve as reference points and aid in generating potential object proposals across the image. Once region proposals are generated by the RPN, they undergo Region of Interest (RoI) pooling or RoIAlign to extract fixed-size feature maps from the CNN feature map [68]. This step aligns the extracted features with the region proposals, enabling subsequent layers to perform accurate object classification and bounding box regression. RoIAlign, in particular, preserves exact spatial locations, avoiding the quantization errors introduced by RoI pooling, which leads to more precise object localization. The RoI features are then fed into a series of fully connected layers, often referred to as the Region-based Convolutional Neural Network (RCNN) head. This head network classifies the object proposals and refines their bounding box coordinates. The classification branch predicts the probability of each proposal belonging to predefined object classes, while the bounding box regression branch adjusts the coordinates of the proposals to better fit the ground truth bounding boxes [69].

During training, Faster R-CNN employs a multi-task loss function that combines classification loss (typically cross-entropy loss) and localization loss (usually smooth L1 loss). The classification loss penalizes incorrect class predictions, while the localization loss penalizes inaccurate bounding box predictions. These losses are computed only for positive RoIs that have sufficient overlap (typically $IoU > 0.5$) with ground truth objects.

To further refine predictions and reduce redundancy, non-maximum suppression (NMS) is applied during inference [70]. NMS suppresses duplicate detections by merging highly overlapping bounding boxes and retaining the box with the highest confidence score.

3.3.6 HRNet Architecture

At its core, HRNet employs a novel architecture that consists of parallel multi-resolution streams and a high-resolution fusion module. The network architecture begins with a basic stem network that processes input images and extracts initial feature maps. From this stem, the architecture diverges into multiple parallel branches, each processing feature maps at different resolutions. These branches operate independently but are later fused using a fusion module that combines information from all scales. This fusion process ensures that the final feature representation maintains both high-level semantic information and fine-grained details essential for accurate object detection.

In the context of object detection, HRNet has been adapted with specific modules such as the HRNet backbone, which is typically pretrained on large-scale image datasets like ImageNet to leverage its feature extraction capabilities [71]. The HRNet backbone extracts multi-scale features efficiently, which are then utilized by subsequent detection heads for tasks such as bounding box prediction and object classification. The multi-scale features are advantageous in detecting objects of various sizes within an image, contributing to improved detection accuracy and robustness. For instance, in a typical object detection pipeline using HRNet, after feature extraction, the network's output is processed by detection heads that perform region proposal and classification tasks. These heads leverage the multi-scale features extracted by HRNet to generate precise object proposals and refine their bounding box coordinates. The integration of HRNet thus enhances the model's ability to localize objects accurately across different scales and achieve superior performance in challenging scenarios [72]. During training, HRNet employs standard optimization techniques such as stochastic gradient descent (SGD) or Adam, coupled with learning rate schedules and batch normalization to stabilize training and accelerate convergence. The network is trained using annotated datasets where ground truth bounding boxes and class labels are provided for supervised learning. Loss functions like cross-entropy loss for classification and smooth L1 loss for bounding box regression are utilized to optimize the network parameters effectively.

3.4 Evaluation Metrices

Evaluation metrics are critical in assessing the performance of machine learning models, particularly in classification and object detection tasks. They provide insights into how well the model is predicting the desired outcomes. The key evaluation metrics used in this study is Mean Average Precision (mAP).

3.4.1 Confusion Matrix

A confusion matrix is a performance measurement tool used to evaluate the effectiveness of a classification model. It provides a comprehensive summary of the prediction results by comparing the actual labels with the predicted labels [73].

To create a confusion matrix, we need four attributes:

True Positives (TP): The model predicted a label and matches correctly as per ground truth.

True Negatives (TN): The model does not predict the label and is not a part of the ground truth.

False Positives (FP): The model predicted a label, but it is not a part of the ground truth (Type

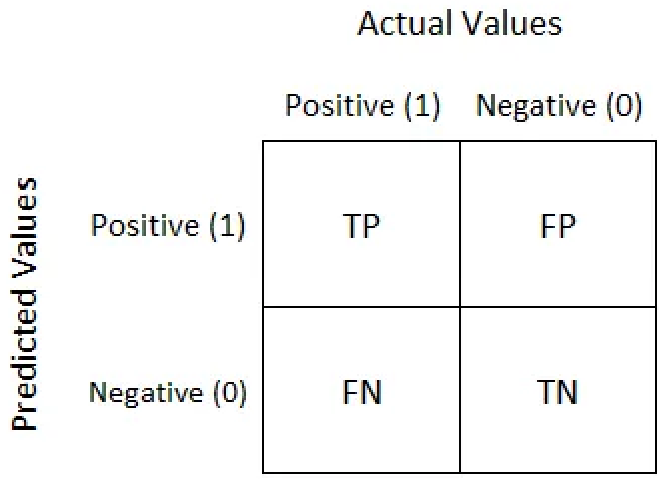


Figure 26 Confusion Matrix

I Error).

False Negatives (FN): The model does not predict a label, but it is part of the ground truth. (Type II Error).

3.4.2 Intersection over Union (IoU)

Intersection over Union indicates the overlap of the predicted bounding box coordinates to the ground truth box. Higher IoU indicates the predicted bounding box coordinates closely resembles the ground truth box coordinates [73].

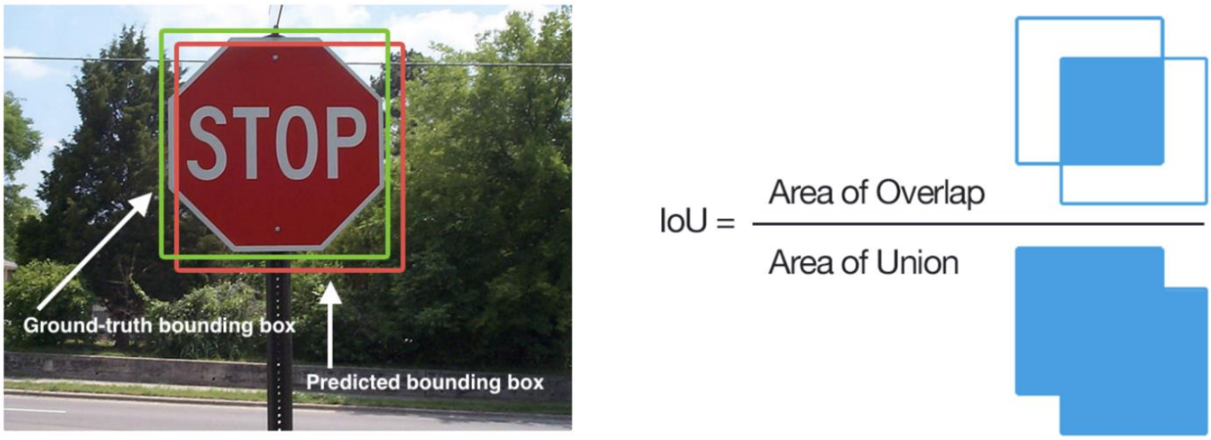


Figure 27 Intersection over Union (IoU) [74]

3.4.3 Precision

Precision measures how well you can find true positives(TP) out of all positive predictions. (TP+FP).

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision is calculated using the IoU threshold in object detection tasks [75]. The precision value may vary based on the model's confidence threshold.

3.4.4 Recall

Recall measures how well you can find true positives(TP) out of all predictions(TP+FN).

$$\text{Recall} = \frac{TP}{TP + FN}$$

3.4.5 Precision-Recall Curve

When it comes to mAP, there is a trade-off between precision and recall. They both consider false positives (FP) and false negatives (FN), making mAP a suitable metric for most detection applications [75].

The Precision-Recall curve is a plot of the precision on the y-axis, and the recall on the x-axis for different thresholds. The curve summarizes the trade-off between the true positive rate and the positive predictive value for a predictive model.

3.4.6 mAP (Mean Average Precision)

The general definition for the Average Precision (AP) is finding the area under the precision-recall curve above.

$$\text{AP} = \int_0^1 p(r)dr$$

Precision and recall are always between 0 and 1. Therefore, AP falls within 0 and 1 also.

The mAP is calculated by finding Average Precision(AP) for each class and then average over a number of classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i$$

The mAP incorporates the trade-off between precision and recall and considers both false positives (FP) and false negatives (FN) [73]. This property makes mAP a suitable metric for most detection applications.

Chapter 4

Results and Discussion

The results of experiments conducted using several cutting-edge object recognition systems are shown and discussed in this section. Finding the model with the highest level of security, accuracy, and computational speed is the primary objective. We experimented with a number of more complex architectures for the final evaluation, including ATSS, RetinaNet, HRNet, YOLOv8, Cascade R-CNN, and Faster R-CNN. Every single one of them had unique machine setups and needs.

Calculating the Mean Average Precision (mAP) with varied Intersection over Union (IoU) criteria is a typical way for determining how well something performs. We employ mAP50, which is the average precision for all categories at a 50% IoU threshold, and mAP50-95, which is the average precision for all categories for a range of IoU thresholds from 50% to 95%. We also consider the number of components in each model, which gives an indicator of the model's size and the quantity of resources required. The rate of pictures processed per second is another metric that indicates how successfully recognition is performed in proportion to the time required. This chapter is meant to present the results of the trials so that you may understand what each object recognition design is and is not capable of. This thorough review also makes it easy to pick the optimum model for a specific application by comparing performance data and resource concerns.

4.1 Performance of Different Baseline Architectures

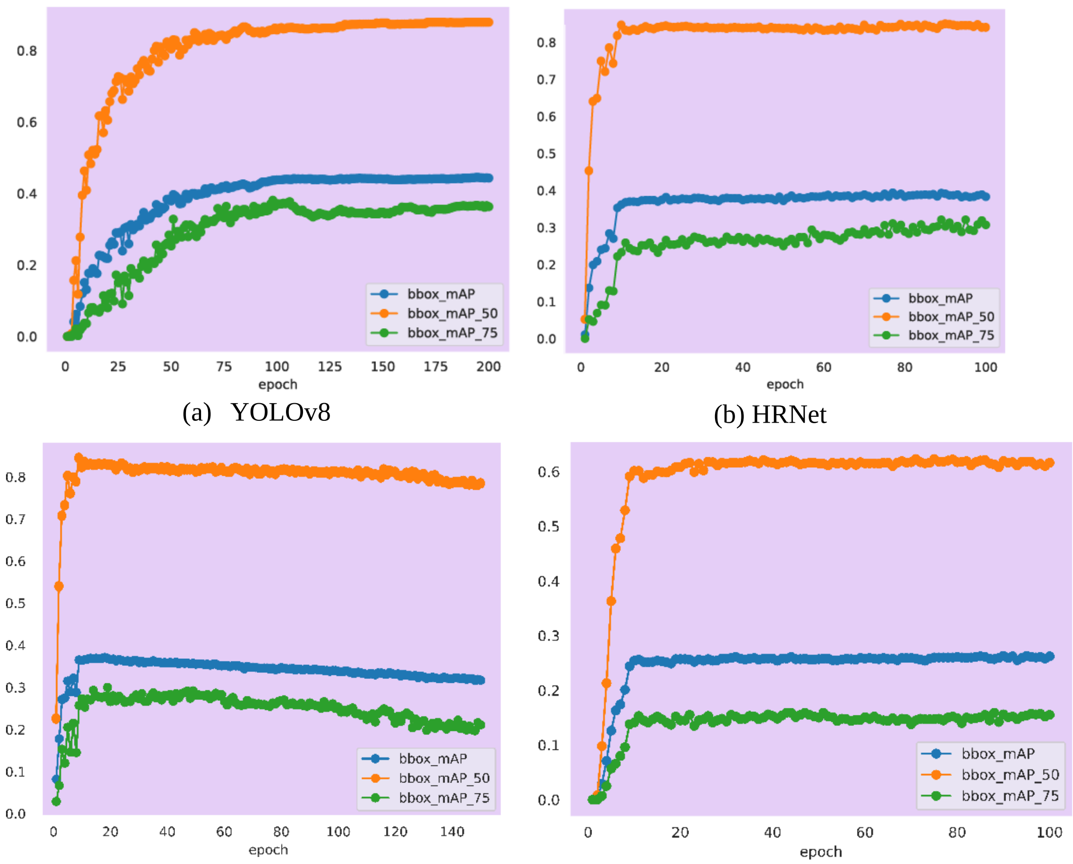
After training each model for 200 epochs with varying parameters, we were able to compare the performance of different object identification architectures. Some of the designs that have been tested are YOLOv8, HRNet, Cascade R-CNN, ATSS, RetinaNet, and Faster R-CNN. Metrics for performance were calculated by averaging the mean absolute precision (mAP) over a variety of intersection over union (IoU) thresholds, with a focus on mAP50 at 50% IoU and mAP50-95 throughout 90% IoU. In addition, we checked the detection frames per second (FPS) and parameter counts of each model. The table below summarises the outcomes of these evaluations:

Table 2 Performances of Different Architectures

Model	mAP50	mAP50-95	Parameter Count (Millions)	Detection speed (images/second)
HRNet	0.905	0.429	46.90	9.2
YOLOv8	0.888	0.470	25.85	16.3
ATSS	0.880	0.418	32.126	14.51
Cascade RCNN	0.840	0.390	6.23	10.53
RetinaNet	0.749	0.318	36.435	15.74
Faster RCNN	0.795	0.347	41.373	13.91

4.2 mAP Curves of Different Architectures

The mAP curve for different architectures representing models' precision-recall for different IoU thresholds. The mAP of Cascade RCNN and ATSS was higher at early epochs and increased further with time while, the mAP of YOLOv8 and HRNet was high at initial epochs and remained high in the subsequent epochs. Although YOLOv8 achieved higher mAPs, Faster RCNN and RetinaNet had lower ones; however, they were trained equally. To do this, recall and precision should be analyzed in detail on mAP curves to determine the relationship between the two for each of the architectures.



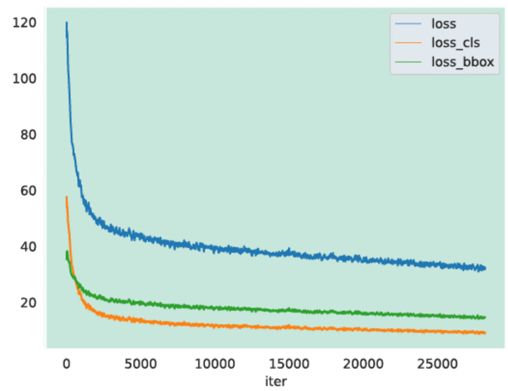
(a) YOLOv8 (b) HRNet

(d) ATSS (c) RetinaNET

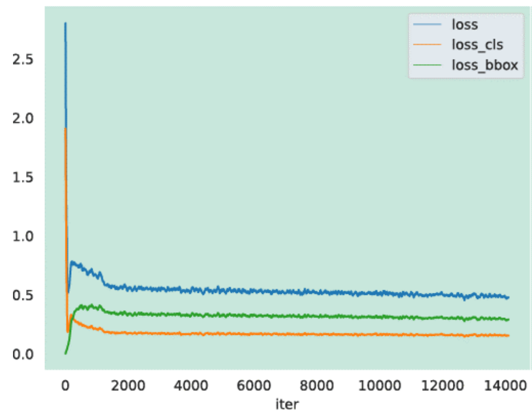
Figure 28 mAP Curves

4.3 Loss Curves of Different Architectures

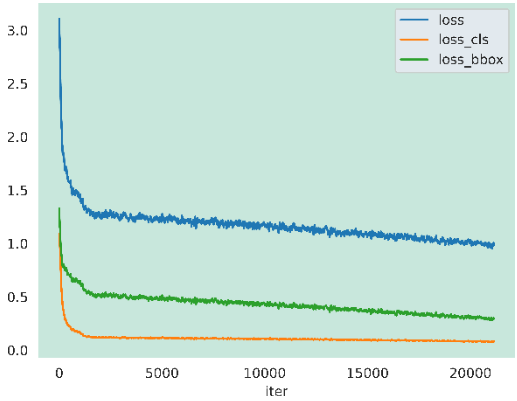
Looking at the loss curves for different designs will help you understand how the training process works and how it converges. Following initial adjustments, YOLOv8 and HRNet demonstrated quick convergence with steady loss levels. Cascade RCNN and AATS converged more slowly but steadily, indicating that they learn slower. RetinaNet and Faster RCNN exhibited larger initial losses, but these losses decreased with time, demonstrating how they learn and how much they can be tweaked. By examining these shapes, you may identify issues with overfitting or underfitting and make the most of your training plan.



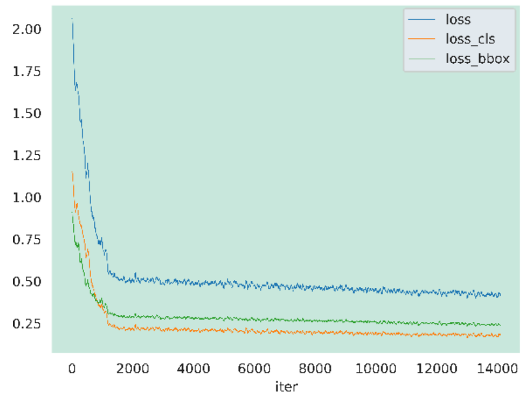
(a) YOLOv8



(b) HRNet



(c) ATSS



(d) RetinaNET

Figure 29 Loss Curves

4.4 Balancing of Class

To improve the model performance and make it more accurate, class imbalance problem had to be solved. The PCB dataset was noted to have a relatively equal distribution of classes in the item counts per image histogram and the annotation heatmap. The number of annotations for each class were as follows: Mouse Bite (267), Spurious Copper (246), Spur (256), Open Circuit (260), Short Circuit (244). The nearly equal distribution of classes in the dataset mean that no single class is dominated and this is important when building models that will be capable of detecting all the forms of errors.

The histogram also shows that the most photos have 6-7 annotations while the number of photos with less or more annotations is quite low. This implies that there is a uniform distribution of imperfections across the sample, which is a good sign. The annotation heatmap also supports the uniformity of the dataset which shows that labels are not clustered in a particular area. The even distribution of data minimizes bias when training the model, therefore enhancing its capability of detecting errors in diverse pictures of PCB. This way, we ensure that our models are well equipped to identify different types of issues while learning from a balanced dataset. This enhances the general performance and reliability of our automated inspection system.

4.5 Qualitative Analysis

To assess the effectiveness of the following object detection models for identifying PCB faults, an analysis was performed. This analysis gave a clear insight on the merits and demerits of each model as seen above.

4.5.1 YOLOv8

This model demonstrated excellent efficiency with a high mAP50 score of 0.888 and a respectable mAP50-95 score of 0.470. Additionally, it could process 16.3 images per second, showcasing its utility in real-time applications. The balance between speed and accuracy makes YOLOv8 particularly suitable for scenarios requiring rapid detection, even if it involves a slight compromise on precision for higher throughput.

4.5.2 HRNet

Achieving the highest mAP50 score of 0.905, HRNet excelled in accuracy. However, its slower mAP50-95 score and image processing rate of 9.2 images per second highlighted its more complex feature extraction approach. HRNet is ideal for applications demanding maximum accuracy, although its slower speed might limit its utility in high-speed inspection lines.

4.5.3 Cascade R-CNN and ATSS

These models showed mAP50 scores of 0.840 and 0.880, respectively, with ATSS also achieving a relatively high recognition speed of 14.51 images per second. These models strike a balance between the speed of YOLOv8 and the accuracy of HRNet. Cascade R-CNN's slower speed and lower parameter efficiency indicate its suitability for complex defect detection tasks where slight gains in accuracy justify longer processing times.

4.5.4 RetinaNet and Faster R-CNN

Both models underperformed in mAP50 and mAP50-95 evaluations. RetinaNet scored a mAP50 of 0.749 and mAP50-95 of 0.318, and its speed and parameter count indicate lower efficiency. Faster R-CNN, with a mAP50 score of 0.795 and a detection speed of 13.91 images per second, showed that it is less effective in terms of both speed and accuracy. These characteristics make RetinaNet and Faster R-CNN less suitable for rapid and accurate PCB defect detection.

4.6 Inference Result

The images provided below show the performance of YOLOv8 Architecture on our PCB defect detection. Figure 30 (a) Annotated Image (Left) (b) Output Image (Right) contains the images of the PCB dataset after Annotation and the Figure 30 (a) Annotated Image (Left) (b) Output Image (Right) shows the Output Image on the right. These figures are important in assessing the effectiveness and deficiencies of various object recognition systems, developed to be used on PCBs.

Observations and Analysis

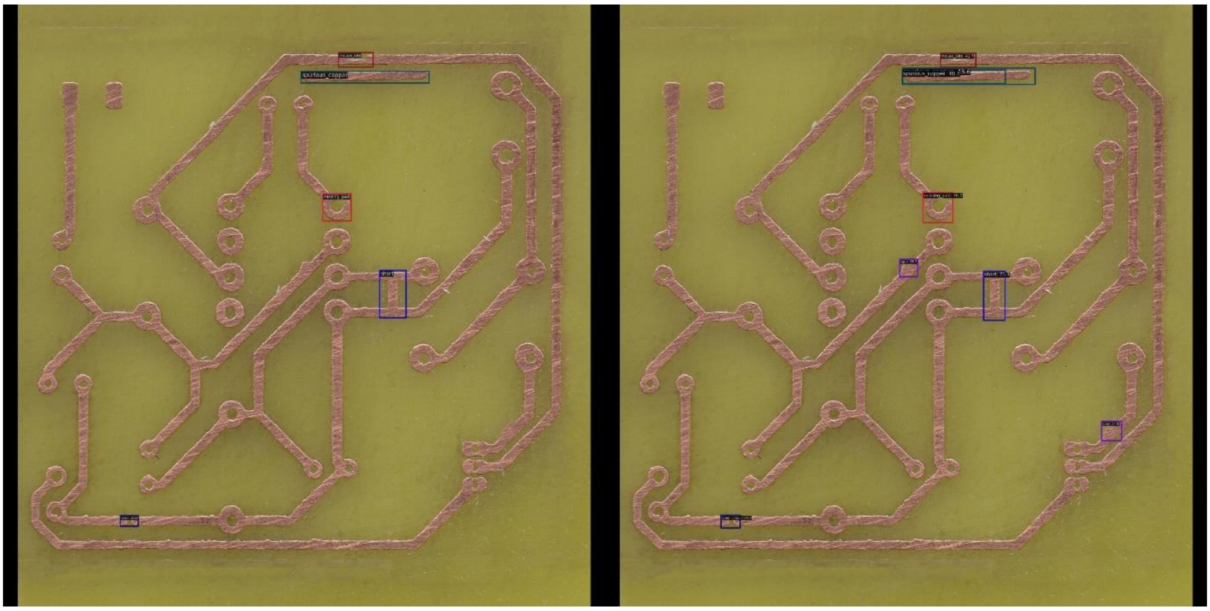


Figure 30 (a) Annotated Image (Left) (b) Output Image (Right)

Despite the promising results shown by the bounding boxes, which are fairly accurate in identifying defects, there remain areas for potential improvement. The main source of the above inaccuracies is in the dataset preparation methodology, more so the chemically etched PCBs. Industry-standard chemically etched PCBs add small features and topographical variations that can affect the accuracy of defect-detection algorithms.

YOLOv8 is the optimal choice for environments where detection speed is critical, owing to its high accuracy and efficiency. While HRNet offers superior accuracy, it is less suited for applications where rapid recognition is essential. Cascade R-CNN and ATSS serve as mid-tier models, providing a reasonable balance between speed and accuracy. The choice of model ultimately depends on the specific requirements of the inspection task, with YOLOv8 being ideal for fast-paced environments and HRNet being suitable for applications where precision is paramount.

Chapter 5

Conclusion

5.1 Summary

This work also provided a comparative analysis of several object detection models with the aim of identifying defects in printed circuit boards (PCBs). The primary concern was to compare the models in terms of the mean Average Precision (mAP) and the time taken for detection. The dataset utilised in this study consisted of six distinct defect categories found in printed circuit boards (PCBs): open circuit, short circuit, spur, mouse bite, missing pad, and spurious copper. The models that were compared are YOLOv8, HRNet, Cascade R-CNN, ATSS, RetinaNet, and Faster R-CNN.

They concluded that YOLOv8 was more accurate and faster in detection than the previous YOLOv8 with an average precision of 0. 888 at 50% intersection over union (mAP50) and the speed of 16. 11 frames per second (FPS). These results make YOLOv8 highly suitable for real-time printed circuit board (PCB) defect identification. HRNet achieved the highest mAP50 of 0. 905, denoting exceptional accuracy. However, its frame rate is 9.2 FPS, which is not suitable for high-speed scenarios due to the low speed. The performance of both ATSS and Cascade R-CNN was found to be balanced, where ATSS achieved an mAP50 of 0. 88 and a processing speed of 14. 51 frames per second (FPS). RetinaNet and Faster R-CNN were considered less suitable for this task due to their lower accuracy and longer time to complete the task.

The study also revealed the importance of preprocessing and data augmentation in enhancing the performance of the model. The techniques applied for improving the model’s robustness and making the distribution of classes in the dataset more equal were scaling, flipping, rotation, shearing, brightness correction, and adding noise.

5.2 Future Work

For future work in the identification of PCB defects, several promising directions can be explored to enhance the object detection models for practicality and effectiveness in the future. One of the most important ones is the normalization of the datasets used for learning and assessment. For the case of PCBs that have been through chemical etching, there are inherent errors on the tracks that could vary depending on the manufacturing processes and layouts.

Through enhanced standardisation measures, it is possible to reduce these differences and aim at attaining better standardisation and homogenisation of the annotations of diverse PCB layouts and types of faults. This process not only enhances the quality of the training data but also increases the model's capability to identify PCB faults even more efficiently, leading to enhanced detection.

Another promising direction for further research is increasing the complexity of the model architecture with the help of advanced techniques such as Controlled Autoregressive (CA) processes. Tuning the parameters of models with high accuracy can significantly improve the speed and accuracy of the models. CA mechanisms include the process of making small changes to model designs in response to feedback, with the aim of improving the ability of the models to correctly detect and classify PCB defects. It can be suggested that researchers enhance the rates of detection and decrease the false positive results in the identification of PCB defects through the adjustment of the models to the specific characteristics of the work. This is particularly relevant for practical use.

Furthermore, the advanced augmentation and synthesis methods can be promising in improving the training sets and robustness of the models. These strategies involve the consideration of other forms of augmentation that deviate from the conventional methods, like synthetic data generation, which introduces more variation and randomness into the training data. Models can be made more versatile for generalisation and to address new issues by exposing them to a wider variety of PCB fault situations that are typical of industrial settings. This method is particularly beneficial when it comes to managing rare or complex fault patterns that may not be adequately captured in small-scale real-world samples.

To ensure that optimised models are practical and effective, more research should be conducted on the application of the models in real-world industries. Evaluating these models in real-life conditions provides insight into their ability to process a large number of samples, stability, and ability to work in automated inspection systems. It is often found that integration into existing production processes can uncover peculiarities and challenges in industrial settings, which can offer opportunities to enhance and fine-tune processes based on real-time feedback and performance evaluations.

Finally, this study creates a solid foundation for improving the identification of PCB defects by employing advanced object detection models. Scholars may further enhance the automation of inspection systems in electronic manufacturing industries by assessing the current limitations and exploring new ideas for enhancement. This will lead to increased reliability, effectiveness, and accuracy of the inspection systems. The constant development of these research lines will be able to offer major advancements in the defect detection technology, which will benefit the manufacturers and the end-users through enhanced quality and productivity of the products.

Chapter 6

OBE Requirements

6.1 Introduction

This project introduces a novel methodology that utilizes Convolutional Neural Networks (CNNs) to detect defects in Printed Circuit Boards (PCBs). Printed circuit boards (PCBs) are crucial elements of contemporary electronic devices, and guaranteeing their impeccable functionality is of utmost importance. This study tackles the aforementioned difficulty by utilizing YOLO-based convolutional neural networks (CNNs) and similar CNNs (e.g., Faster RNN, R-FCN etc.) to effectively and precisely detect various flaws, including but not limited to missing holes, open circuits, and short circuit.

The accuracy and effectiveness of traditional flaw detection methods in printed circuit boards (PCBs) are frequently constrained. In order to address these limitations, the present study proposes YOLO CNNs, a cutting-edge deep learning methodology, to automate the detection procedure. You-Only-Look-Once (YOLO) Convolutional Neural Networks (CNNs) demonstrate an impressive ability to efficiently analyze whole Printed Circuit Board (PCB) pictures in real-time, facilitating thorough detection of defects with exceptional precision and efficiency.

6.2 Course Outcomes (COs) Addressed

The following table shows the COs for EEE 4700/4800 (Project and Thesis).

COs	CO Statement	POs
CO1	Identify a contemporary real-life problem related to electrical and electronic engineering by reviewing and analyzing existing research works.	PO2
CO2	Determine functional requirements of the problem considering feasibility and efficiency through analysis and synthesis of information.	PO4
CO3	Select a suitable solution and determine its method considering professional ethics, codes and standards.	PO8
CO4	Adopt modern engineering resources and tools for the solution of the problem.	PO5
CO5	Prepare management plan and budgetary implications for the solution of the problem.	PO11
CO6	Analyze the impact of the proposed solution on health, safety, culture and society.	PO6
CO7	Analyze the impact of the proposed solution on environment and sustainability.	PO7
CO8	Develop a viable solution considering health, safety, cultural, societal and environmental aspects.	PO3
CO9	Work effectively as an individual and as a team member for the accomplishment of the solution.	PO9
CO10	Prepare various technical reports, design documentation, and deliver effective presentations for demonstration of the solution.	PO10
CO11	Recognize the need for continuing education and participation in professional societies and meetings.	PO12

Table 3 Course Outcomes (COs) Addressed

The following table explains or justifies how the COs and corresponding POs have been addressed in EEE 4700/4800 (Project and Thesis).

COs	POs	Explanation/Justification
CO1	PO2	
CO2	PO4	Conducting investigations into complex electrical engineering issues related to PCB defect detection. We design experiments to collect data on various PCB flaws, analyze and interpret this data using YOLO models, and synthesize information to draw valid conclusions about defect identification. Through rigorous experimentation, data analysis, and information synthesis, our project provides insights into effective defect detection methodologies, contributing to the advancement of PCB manufacturing processes and product quality. Our project primarily covers the safety and societal aspects of PO3. Safety is addressed through the detection of defects in PCBs, ensuring that electronic devices built with these boards meet safety standards and do not pose risks to users. Societal considerations are addressed by focusing on the growing demand for electronic devices and the importance of reliable PCBs in meeting this demand, thereby contributing to consumer satisfaction and confidence in electronic products. While the project indirectly impacts public health by ensuring safer electronic devices, it does not directly address cultural or environmental aspects.
CO3	PO8	The development of PCB defect detection systems adheres to ethical guidelines, such as prioritizing the safety and reliability of electronic devices.
CO4	PO5	
CO5	PO11	In managing our project, we opted to source industrial-grade products from Old Dhaka instead of lab-grade ones, showcasing our adeptness in project management and economic decision-making. By leveraging local resources, we ensured cost-effectiveness without compromising on

		quality, a testament to our understanding of engineering management principles.
CO6	PO6	Our project addresses the societal facet of PO6 by considering the societal demand for reliable electronic devices and the implications of PCB defects on consumer satisfaction. Additionally, it touches on the safety facet by ensuring that defect detection in PCBs contributes to the safety and reliability of electronic products, aligning with professional engineering practice standards.
CO7	PO7	Our project's contribution to societal aspects lies in enhancing the reliability of electronic devices through improved PCB defect detection, thereby meeting societal expectations for safer and more dependable technology.
CO8	PO3	Our project primarily covers the safety and societal aspects. Safety is addressed through the detection of defects in PCBs, ensuring that electronic devices built with these boards meet safety standards and do not pose risks to users. Societal considerations are addressed by focusing on the growing demand for electronic devices and the importance of reliable PCBs in meeting this demand, thereby contributing to consumer satisfaction and confidence in electronic products. While the project indirectly impacts public health by ensuring safer electronic devices, it does not directly address cultural or environmental aspects.
CO9	PO9	Our project requires both individual expertise, such as in designing experiments and analyzing data, and effective teamwork to collaborate on implementing advanced image processing models and developing comprehensive defect detection solutions.
CO10	PO10	By comprehensively documenting our research process, findings, and methodologies in detailed reports. We also create design documentation outlining our defect detection system, and make effective presentations to communicate our results to both the engineering community and society.
CO11	PO12	

Table 4 Explanations/Justifications on how the COs and corresponding POs have been addressed

6.3 Aspects of Program Outcomes (POs) Addressed

The following table shows the aspects addressed for certain Program Outcomes (POs) addressed in EEE 4700/4800 for Project and Thesis.

	Statement	Different Aspects	Put Tick (√)
PO3	**Design/development of solutions**: Design solutions for complex electrical and electronic engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	Public health	
		Safety	√
		Cultural	
		Societal	√
		Environmental	
PO4	**Investigation:** Conduct investigations of complex electrical and electronic engineering problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.	Design of experiments	√
		Analysis and interpretation of data	√
		Synthesis of information	√
PO6	**The engineer and society:** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice and solutions to complex electrical and electronic engineering problems.	Societal	√
		Health	
		Safety	√
		Legal	
		Cultural	
PO7	**Environment and sustainability:** Understand and evaluate the sustainability and impact of professional engineering work in the solution of complex electrical and	Societal	√
		Environmental	

	electronic engineering problems in societal and environmental contexts.		
PO8	**Ethics:** Apply ethical principles embedded with religious values, professional ethics and responsibilities, and norms of electrical and electronic engineering practice.	Religious values	
		Professional ethics and responsibilities	√
		Norms	√
PO9	**Individual work and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.	Individual	√
		Teamwork	√
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	Comprehend and write effective reports	√
		Design documentation	√
		Make effective presentations	√
		Give and receive clear instructions	√
PO11	**Project management and finance:** Demonstrate knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	Engineering management principles	√
		Economic decision-making	√
		Manage projects	√
		Multidisciplinary environments	

Table 5 Program Outcomes (POs) addressed in EEE 4700/4800 (Project and Thesis)

6.4 Addressing Knowledge Profiles (K3 – K8)

The following table shows the Knowledge Profiles (K3 – K8) addressed in EEE 4700/4800 (Project and Thesis).

K	Knowledge Profile (Attribute)	Put Tick (√)
K3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline	
K4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline	√
K5	Knowledge that supports engineering design in a practice area	√
K6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline	√
K7	Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the engineer's professional responsibility to public safety; the impacts of engineering activity; economic, social, cultural, environmental and sustainability	√
K8	Engagement with selected knowledge in the research literature of the discipline	√

Table 6 Knowledge Profiles (K3 – K8) addressed in EEE 4700/4800 (Project and Thesis)

The following table explains or justifies how the Knowledge Profiles (K3 – K8) have been addressed in EEE 4700/4800 (Project and Thesis).

K	Explanation/Justification
K3	
K4	Machine learning & Computer vision techniques will be used which contain theoretical framework as the accepted practice areas in the engineering discipline. By incorporating advanced Convolutional Neural Networks (CNNs) like YOLO-based architectures, we utilize cutting-edge techniques to detect defects in PCBs.
K5	Designing our Computer Vision frameworks extends to the practice area of PCB manufacturing in the industry setting.
K6	By integrating knowledge of engineering practice, specifically in technology related to the field of printed circuit boards (PCBs). Through the implementation of advanced

	Convolutional Neural Networks (CNNs) and computer vision techniques, we apply cutting-edge technology to detect defects in PCBs with precision and efficiency.
K7	We prioritize ethics and professional responsibility by ensuring the safety and reliability of electronic devices through accurate defect detection. Additionally, our project considers the economic, social, and environmental impacts of PCB manufacturing, contributing to sustainability and societal well-being.
K8	We extensively review existing literature on computer vision, deep learning, and PCB manufacturing to inform our approach. By incorporating findings from relevant studies and staying abreast of advancements in the field, we ensure that our project is grounded in established knowledge and benefits from insights gained from prior research.

Table 7 Explanations/Justifications on how the Knowledge Profiles (K3 – K8) have been addressed

6.5 Addressing Attributes of Ranges of Complex Engineering Problem Solving (P1 – P7)

The following table shows the attributes of ranges of Complex Engineering Problem Solving (P1 – P7) addressed in EEE 4700/4800 (Project and Thesis).

P	**Range of Complex Engineering Problem Solving**	**Put Tick (√)**
Attribute	Complex Engineering Problems have characteristic P1 and some or all of P2 to P7:	
Depth of knowledge required	**P1:** Cannot be resolved without in-depth engineering knowledge at the level of one or more of K3, K4, K5, K6 or K8 which allows a fundamentals-based, first principles analytical approach	√
Range of conflicting requirements	**P2:** Involve wide-ranging or conflicting technical, engineering and other issues	√
Depth of analysis required	**P3:** Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models	√
Familiarity of issues	**P4:** Involve infrequently encountered issues	
Extent of applicable codes	**P5:** Are outside problems encompassed by standards and codes of practice for professional engineering	
Extent of stakeholder involvement and conflicting requirements	**P6:** Involve diverse groups of stakeholders with widely varying needs	
Interdependence	**P7:** Are high level problems including many component parts or sub-problems	√

Table 8 Attributes of ranges of Complex Engineering Problem Solving (P1 – P7) addressed in EEE 4700/4800 (Project and Thesis)

The following table explains or justifies how the attributes of ranges of Complex Engineering Problem Solving (P1 – P7) have been addressed in EEE 4700/4800 (Project and Thesis).

P	**Explanation/Justification**
P1	We will need in-depth engineering knowledge along with K4, K5, K6, K7 and K8
P2	Balancing real-time output with accuracy is critical in our project, which aims to accomplish automated, realtime defect identification inside manufacturing settings.

P3	Identifying diverse faults demands abstract thinking to build analytical models that have multiple approaches to begin with.
P4	
P5	
P6	
P7	Involves a variety of interconnected complications, necessitating evaluation of various component elements or sub-problems within the detection process.

Table 9 Explanations/Justifications on how the attributes of ranges of Complex Engineering Problem Solving (P1 – P7) have been addressed

6.6 Addressing Attributes of Ranges of Complex Engineering Activities (A1-A5)

The following table shows the attributes of ranges of Complex Engineering Activities (A1 – A5) addressed in EEE 4700/4800 (Project and Thesis).

A	**Range of Complex Engineering Activities**	**Put Tick (√)**
Attribute	Complex activities means (engineering) activities or projects that have some or all of the following characteristics:	
Range of resources	**A1:** Involve the use of diverse resources (and for this purpose resources include people, money, equipment, materials, information and technologies)	√
Level of interaction	**A2:** Require resolution of significant problems arising from interactions between wide-ranging or conflicting technical, engineering or other issues	√
Innovation	**A3:** Involve creative use of engineering principles and research-based knowledge in novel ways	√
Consequences for society and the environment	**A4:** Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation	√
Familiarity	**A5:** Can extend beyond previous experiences by applying principles-based approaches	

Table 10 Attributes of ranges of Complex Engineering Activities (A1 – A5) addressed in EEE 4700/4800 (Project and Thesis)

The following table explains or justifies how the attributes of ranges of Complex Engineering Activities (A1 – A5) have been addressed in EEE 4700/4800 (Project and Thesis).

A	Explanation/Justification
A1	Our project involves diverse resources such as advanced technologies (Convolutional Neural Networks), expert knowledge, equipment for data collection, and materials for experimentation, enabling comprehensive solutions to complex PCB defect detection problems. In addition to advanced technologies and expert knowledge, our project utilizes a range of materials like Isopropanol and CCB single-sided boards.
A2	Our project navigates significant challenges stemming from various technical aspects, including optimizing CNN architectures, integrating data collection equipment, and balancing conflicting requirements like accuracy versus latency in our model, highlighting complex engineering problem-solving.
A3	In addition to optimizing PCB manufacturing for enhanced training data, we creatively implemented data augmentation techniques to diversify our dataset. We also experimented with novel CNN architectures tailored specifically for PCB defect detection. These creative approaches demonstrate innovative problem-solving and push the boundaries of traditional methods in our project.
A4	Our project's significance in PCB defect detection extends beyond technical realms, impacting societal and environmental contexts. Predicting and mitigating consequences, such as ensuring electronic device reliability and minimizing electronic waste from faulty PCBs, pose challenges. Addressing these complexities underscores the multifaceted nature of our engineering problem.
A5	

Table 11 Explanations/Justifications on how the attributes of ranges of Complex Engineering Activities (A1 – A5) have been addressed in EEE 4700/4800 (Project and Thesis).

6.7 Socio-cultural, environmental, and ethical impact of the project

Socio-cultural: The project significantly influences the discourse surrounding the technological prowess of Bangladesh within a socio-cultural framework. In a socio-cultural context, the project plays a crucial role in shaping the narrative of technological capabilities within Bangladesh. As Walton Digi-Tech Industries Limited gains prominence in PCB production and export, the research offers a unique opportunity to showcase the nation's growing expertise in an industry critical to modern technology.

The rise of 'Made in Bangladesh' printed circuit boards (PCBs) and their successful export to Greece by Walton Digi-Tech Industries Limited is not merely a business achievement; it serves as a symbol of national pride and accomplishment. The project, by focusing on local PCBs and their flaws, not only advances the technological landscape but also simplifies the process of determining faults in PCBs.

Environmental: PCBs that have been identified as defective in conventional manufacturing processes are frequently disposed of, resulting in substantial electronic refuse and environmental complications. The project's premeditated introduction of defects throughout the manufacturing process can be regarded as a strategic and environmentally conscious measure. Instead of exacerbating the escalating issue of electronic waste, these defective printed circuit boards transform into valuable assets in the pursuit of improving fault detection algorithms. The transition to automated fault detection not only improves the precision and effectiveness of the manufacturing procedure but also fosters a more environmentally conscious and sustainable approach to printed circuit board (PCB) production.

Ethical: An emphasis is placed on product quality and customer fulfillment in this endeavor. The research guarantees the creation of dependable and precise fault detection algorithms by deliberately introducing defects and methodically structuring a dataset that virtually represents printed circuit boards from Bangladesh. This dedication to precision and dependability is consistent with manufacturing ethics, which promotes industry-wide accountability and transparency. The project's objective of improving fault identification in printed circuit boards (PCBs) manufactured in Bangladesh indirectly aids in the manufacturing of dependable and superior electronics. This, in turn, mitigates the occurrence of defective products in the market and, as a result, potential ethical dilemmas associated with consumer discontentment or safety.

References

[1] J. LaDou, “Printed circuit board industry,” *Int J Hyg Environ Health*, vol. 209, no. 3, pp. 211–219, May 2006, doi: 10.1016/J.IJHEH.2006.02.001.

[2] V. U. Sankar, G. Lakshmi, and Y. S. Sankar, “A Review of Various Defects in PCB,” *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 38, no. 5, pp. 481–491, Oct. 2022, doi: 10.1007/S10836-022-06026-7/FIGURES/3.

[3] B. Hu and J. Wang, “Detection of PCB Surface Defects with Improved Faster-RCNN and Feature Pyramid Network,” *IEEE Access*, vol. 8, pp. 108335–108345, 2020, doi: 10.1109/ACCESS.2020.3001349.

[4] L. Song *et al.*, “PCB Electronic Component Defect Detection Method based on Improved YOLOv4 Algorithm,” *J Phys Conf Ser*, vol. 1827, no. 1, p. 012167, Mar. 2021, doi: 10.1088/1742-6596/1827/1/012167.

[5] S. Tang, F. He, X. Huang, and J. Yang, “Online PCB Defect Detector On A New PCB Defect Dataset,” Feb. 2019, Accessed: Jun. 25, 2024. [Online]. Available: https://arxiv.org/abs/1902.06197v1

[6] B. Du, F. Wan, G. Lei, L. Xu, C. Xu, and Y. Xiong, “YOLO-MBBi: PCB Surface Defect Detection Method Based on Enhanced YOLOv5,” *Electronics 2023, Vol. 12, Page 2821*, vol. 12, no. 13, p. 2821, Jun. 2023, doi: 10.3390/ELECTRONICS12132821.

[7] J. Tang, S. Liu, D. Zhao, L. Tang, W. Zou, and B. Zheng, “PCB-YOLO: An Improved Detection Algorithm of PCB Surface Defects Based on YOLOv5,” *Sustainability 2023, Vol. 15, Page 5963*, vol. 15, no. 7, p. 5963, Mar. 2023, doi: 10.3390/SU15075963.

[8] C. Liu, X. Zhou, J. Li, and C. Ran, “PCB Board Defect Detection Method based on Improved YOLOv8,” *Frontiers in Computing and Intelligent Systems*, vol. 6, no. 2, pp. 1–6, Dec. 2023, doi: 10.54097/FCIS.V6I2.01.

[9] H. Aregawi and M. Abdo, “Effects of etching process inaccuracy in the malfunctioning level of PCB circuits - a simulaton based analysis,” *Zede Journal*, vol. 38, no. 1, pp. 53–63, Dec. 2020, Accessed: Jun. 25, 2024. [Online]. Available: https://www.ajol.info/index.php/zj/article/view/202431

[10] M. Yu, X. Zeng, Q. Song, L. Liu, and J. Li, “Examining regeneration technologies for etching solutions: a critical analysis of the characteristics and potentials,” *J Clean Prod*, vol. 113, pp. 973–980, Feb. 2016, doi: 10.1016/J.JCLEPRO.2015.10.131.

[11] Y. Bi, B. Xue, P. Mesejo, S. Cagnoni, and M. Zhang, “A Survey on Evolutionary Computation for Computer Vision and Image Analysis: Past, Present, and Future Trends,” *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 1, pp. 5–25, Feb. 2023, doi: 10.1109/TEVC.2022.3220747.

[12] T. K. Gupta and K. Raza, “Optimization of ANN Architecture: A Review on Nature-Inspired Techniques,” *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging*, pp. 159–182, Jan. 2019, doi: 10.1016/B978-0-12-816086-2.00007-2.

[13] J.-W. Lin, “Artificial Neural Network Related to Biological Neuron Network: A Review,” *Advanced Studies in Medical Sciences*, vol. 5, no. 1, pp. 55–62, 2017, doi: 10.12988/asms.2017.753.

[14] Z. Jin, “Application of WCA-RBF Neural Network in Fault Diagnosis of Analog Circuits,” *International Transactions on Electrical Energy Systems*, vol. 2023, 2023, doi: 10.1155/2023/8812152.

[15] Engin Pekel, “A COMPREHENSIVE REVIEW FOR ARTIFICAL NEURAL NETWORK APPLICATION TO PUBLIC TRANSPORTATION,” Hitit University. Accessed: Jun. 24, 2024. [Online]. Available: https://www.researchgate.net/publication/315111480_A_COMPREHENSIVE_REVIEW_FOR_ARTIFICAL_NEURAL_NETWORK_APPLICATION_TO_PUBLIC_TRANSPORTATION

[16] J. Singh and R. Banerjee, “A study on single and multi-layer perceptron neural network,” *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, pp. 35–40, Mar. 2019, doi: 10.1109/ICCMC.2019.8819775.

[17] Shekhar Banerjee, “Exploring the Power and Limitations of Multi-Layer Perceptron (MLP) in Machine Learning,” Medium. Accessed: Jun. 24, 2024. [Online]. Available: https://shekhar-banerjee96.medium.com/exploring-the-power-and-limitations-of-multi-layer-perceptron-mlp-in-machine-learning-d97a3f84f9f4

[18] M.-C. Popescu and V. E. Balas, “Multilayer Perceptron and Neural Networks”.

[19] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 943–947, Nov. 2015, doi: 10.22214/ijraset.2022.47789.

[20] M. Azzeh, A. Alkhateeb, and A. Bou Nassif, “Software effort estimation using convolutional neural network and fuzzy clustering,” *Neural Comput Appl*, 2024, doi: 10.1007/S00521-024-09855-Z.

[21] M. J. Akhtar *et al.*, “A Robust Framework for Object Detection in a Traffic Surveillance System,” *Electronics (Switzerland)*, vol. 11, no. 21, Nov. 2022, doi: 10.3390/ELECTRONICS11213425.

[22] R. S. Samosir, E. Abdurachman, F. L. Gaol, and B. S. Sabarguna, “Hybrid method architecture design of mri brain tumors image segmentation,” *ICIC Express Letters*, vol. 14, no. 12, pp. 1177–1184, Dec. 2020, doi: 10.24507/ICICEL.14.12.1177.

[23] “Convolutional Neural Network and its Architectures,” 2021, doi: 10.37591/JoCTA.

[24] M. A. Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. D. B. Bruce, “Position, Padding and Predictions: A Deeper Look at Position Information in CNNs,” *Int J Comput Vis*, pp. 1–22, Apr. 2024, doi: 10.1007/S11263-024-02069-9/FIGURES/18.

[25] S. K. R. Meruva, V. G. S. Tulasi, N. Vinnakota, and V. Bhavana, “Risk Level Prediction of Diabetic Retinopathy based on Retinal Images using Deep Learning Algorithm,” *Procedia Comput Sci*, vol. 215, pp. 722–730, 2022, doi: 10.1016/J.PROCS.2022.12.074.

[26] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[27] P. Rajeshwari, P. Abhishek, and P. S. | T. Vinod, “Object Detection: An Overview,” *International Journal of Trend in Scientific Research and Development*, vol. Volume-3, no. Issue-3, pp. 1663–1665, Apr. 2019, doi: 10.31142/IJTSRD23422.

[28] M. V. Athira and D. M. Khan, “Recent Trends on Object Detection and Image Classification: A Review,” *2020 International Conference on Computational Performance Evaluation, ComPE 2020*, pp. 427–435, Jul. 2020, doi: 10.1109/COMPE49325.2020.9200080.

[29] C. Li *et al.*, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” Sep. 2022, Accessed: Jun. 24, 2024. [Online]. Available: https://arxiv.org/abs/2209.02976v1

[30] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/J.PATCOG.2017.10.013.

[31] Y. Chen, M. C. Goorden, F. J. Beekman, L. Du, R. Zhang, and X. Wang, “Overview of two-stage object detection algorithms,” *J Phys Conf Ser*, vol. 1544, no. 1, p. 012033, May 2020, doi: 10.1088/1742-6596/1544/1/012033.

[32] M. Asif, S. Liu, G. M. Ali, A. Bouferguene, and M. Al-Hussein, “The Effectiveness of Data Augmentation in Construction Site-Related Image Classification,” *Lecture Notes in Civil Engineering*, vol. 363 LNCE, pp. 247–257, 2023, doi: 10.1007/978-3-031-34593-7_16.

[33] “Closer Look at IC Pin Inspection With Bi-Telecentric Lenses,” *https://vicoimaging.com/*, Accessed: Jun. 24, 2024. [Online]. Available: https://vicoimaging.com/bi-telecentric-lenses-for-ic-pin-inspection/

[34] S. D. Kalro, P. B. G, M. B. S, and P. H. D, “PCB Defect Detection Using Image Subtraction Algorithm,” *International Journal of Computer Science Trends and Technology*, vol. 3, Accessed: Jun. 24, 2024. [Online]. Available: www.ijcstjournal.org

[35] R. R. Chavan, S. A. Chavan, G. D. Dokhe, M. B. Wagh, and A. S. Vaidya, “Quality Control of PCB using Image Processing,” *Article in International Journal of Computer Applications*, vol. 141, no. 5, pp. 975–8887, 2016, doi: 10.5120/ijca2016909623.

[36] V. Chaudhary, I. R. Dave, and K. P. Upla, “Automatic visual inspection of printed circuit board for defect detection and classification,” *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, vol. 2018-January, pp. 732–737, Jul. 2017, doi: 10.1109/WISPNET.2017.8299858.

[37] “Venture’s PCB Assembly Visual Inspection for Quality Control.” Accessed: Jun. 24, 2024. [Online]. Available: https://www.venture-mfg.com/visual-quality-inspection/

[38] V. H. Gaidhane, Y. V. Hote, and V. Singh, “An efficient similarity measure approach for PCB surface defect detection,” *Pattern Analysis and Applications*, vol. 21, no. 1, pp. 277–289, Feb. 2018, doi: 10.1007/S10044-017-0640-9/TABLES/5.

[39] D. M. Tsai and C. K. Huang, “Defect Detection in Electronic Surfaces Using Template-Based Fourier Image Reconstruction,” *IEEE Trans Compon Packaging Manuf Technol*, vol. 9, no. 1, pp. 163–172, Jan. 2019, doi: 10.1109/TCPMT.2018.2873744.

[40] A. A. I. M. Hassanin, F. E. Abd El-Samie, and G. M. El Banby, “A real-time approach for automatic defect detection from PCBs based on SURF features and morphological operations,” *Multimed Tools Appl*, vol. 78, no. 24, pp. 34437–34457, Dec. 2019, doi: 10.1007/S11042-019-08097-9/TABLES/3.

[41] Z. Yuan, X. Tang, H. Ning, and Z. Yang, “LW-YOLO: Lightweight Deep Learning Model for Fast and Precise Defect Detection in Printed Circuit Boards,” *Symmetry 2024, Vol. 16, Page 418*, vol. 16, no. 4, p. 418, Apr. 2024, doi: 10.3390/SYM16040418.

[42] K. D. Kang, I. M. S. K. Ilankoon, N. Dushyantha, and M. N. Chong, “Assessment of Pre-Treatment Techniques for Coarse Printed Circuit Boards (PCBs) Recycling,” *Minerals 2021, Vol. 11, Page 1134*, vol. 11, no. 10, p. 1134, Oct. 2021, doi: 10.3390/MIN11101134.

[43] A. Bhattacharya and S. G. Cloutier, “End-to-end deep learning framework for printed circuit board manufacturing defect classification,” *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1–13, Jul. 2022, doi: 10.1038/s41598-022-16302-3.

[44] K. Singh, S. Kharche, A. Chauhan, and P. Salvi, “PCB Defect Detection Methods: A Review of Existing Methods and Potential Enhancements,” *Journal of Engineering Science and Technology Review*, vol. 17, no. 1, pp. 156–167, 2024, doi: 10.25103/jestr.171.19.

[45] G. Liu and H. Wen, “Printed circuit board defect detection based on MobileNet-Yolo-Fast,” *https://doi.org/10.1117/1.JEI.30.4.043004*, vol. 30, no. 4, p. 043004, Jul. 2021, doi: 10.1117/1.JEI.30.4.043004.

[46] Z. Xiong, “A Design of Bare Printed Circuit Board Defect Detection System Based on YOLOv8,” *Highlights in Science, Engineering and Technology*, vol. 57, pp. 203–209, Jul. 2023, doi: 10.54097/HSET.V57I.10002.

[47] W. Huang and P. Wei, “A PCB Dataset for Defects Detection and Classification,” vol. 14, no. 8, Jan. 2019, Accessed: Jun. 24, 2024. [Online]. Available: https://arxiv.org/abs/1901.08204v1

[48] V. A. Adibhatla, H. C. Chih, C. C. Hsu, J. Cheng, M. F. Abbod, and J. S. Shieh, “Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once,” *Mathematical Biosciences and Engineering*, vol. 18, no. 4, pp. 4411–4428, 2021, doi: 10.3934/MBE.2021223.

[49] V. A. Adibhatla, H. C. Chih, C. C. Hsu, J. Cheng, M. F. Abbod, and J. S. Shieh, “Defect Detection in Printed Circuit Boards Using You-Only-Look-Once Convolutional Neural Networks,” *Electronics 2020, Vol. 9, Page 1547*, vol. 9, no. 9, p. 1547, Sep. 2020, doi: 10.3390/ELECTRONICS9091547.

[50] W. Chen, Z. Huang, Q. Mu, and Y. Sun, "PCB Defect Detection Method Based on Transformer-YOLO," *IEEE Access*, vol. 10, pp. 129480–129489, 2022, doi: 10.1109/ACCESS.2022.3228206.

[51] G. Zhou, L. Yu, Y. Su, B. Xu, and G. Zhou, "Lightweight PCB defect detection algorithm based on MSD-YOLO," *Cluster Comput*, vol. 27, no. 3, pp. 3559–3573, Jun. 2023, doi: 10.1007/S10586-023-04156-X/FIGURES/15.

[52] K. Xia *et al.*, "Global contextual attention augmented YOLO with ConvMixer prediction heads for PCB surface defect detection," *Scientific Reports 2023 13:1*, vol. 13, no. 1, pp. 1–16, Jun. 2023, doi: 10.1038/s41598-023-36854-2.

[53] Y. ; Jiang, M. ; Cai, D. Zhang, Y. Jiang, M. Cai, and D. Zhang, "Lightweight Network DCR-YOLO for Surface Defect Detection on Printed Circuit Boards," *Sensors 2023, Vol. 23, Page 7310*, vol. 23, no. 17, p. 7310, Aug. 2023, doi: 10.3390/S23177310.

[54] F. R. Leta, F. F. Feliciano, and F. P. R. Martins, "COMPUTER VISION SYSTEM FOR PRINTED CIRCUIT BOARD INSPECTION".

[55] Q. Ling and N. A. M. Isa, "Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey," *IEEE Access*, vol. 11, pp. 15921–15944, 2023, doi: 10.1109/ACCESS.2023.3245093.

[56] P. Chen and F. Xie, "A Machine Learning Approach for Automated Detection of Critical PCB Flaws in Optical Sensing Systems," *Photonics 2023, Vol. 10, Page 984*, vol. 10, no. 9, p. 984, Aug. 2023, doi: 10.3390/PHOTONICS10090984.

[57] B. Feng and J. Cai, "PCB Defect Detection via Local Detail and Global Dependency Information," *Sensors 2023, Vol. 23, Page 7755*, vol. 23, no. 18, p. 7755, Sep. 2023, doi: 10.3390/S23187755.

[58] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/S11042-022-13644-Y/TABLES/7.

[59] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection," *Inf Sci (N Y)*, vol. 522, pp. 241–258, Jun. 2020, doi: 10.1016/J.INS.2020.02.067.

[60] D.-H. Jeon, T.-S. Kim, and J.-S. Kim, "A Method for Reducing False Negative Rate in Non-Maximum Suppression of YOLO Using Bounding Box Density," *Journal of Multimedia Information System*, vol. 10, no. 4, pp. 293–300, Dec. 2023, doi: 10.33851/JMIS.2023.10.4.293.

[61] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video," *Journal of Computational Vision and Imaging Systems*, vol. 3, no. 1, Sep. 2017, doi: 10.15353/vsnl.v3i1.171.

[62] Y. Li and F. Ren, "Light-Weight RetinaNet for Object Detection," May 2019, Accessed: Jun. 24, 2024. [Online]. Available: https://arxiv.org/abs/1905.10011v1

[63] X. Li, H. Zhao, and L. Zhang, “Recurrent RetinaNet: A Video Object Detection Model Based on Focal Loss,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11304 LNCS, pp. 499–508, 2018, doi: 10.1007/978-3-030-04212-7_44.

[64] L. J. Biffi *et al.*, “ATSS Deep Learning-Based Approach to Detect Apple Fruits,” *Remote Sensing 2021, Vol. 13, Page 54*, vol. 13, no. 1, p. 54, Dec. 2020, doi: 10.3390/RS13010054.

[65] L. Shi, H. Zu, J. Tai, and W. Niu, “A welding defect detection model based on a shape-aware network,” *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 5, pp. 6147–6162, Jan. 2022, doi: 10.3233/JIFS-220132.

[66] G. Ghiasi, T.-Y. Lin, and Q. V Le Google Brain, “NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection.” pp. 7036–7045, 2019.

[67] X. Chen and A. Gupta, “An Implementation of Faster RCNN with Study for Region Sampling,” Feb. 2017, Accessed: Jun. 24, 2024. [Online]. Available: https://arxiv.org/abs/1702.02138v2

[68] R. Gavrilescu, C. Zet, C. Fosalau, M. Skoczylas, and D. Cotovanu, “Faster R-CNN:an Approach to Real-Time Object Detection,” *EPE 2018 - Proceedings of the 2018 10th International Conference and Expositions on Electrical And Power Engineering*, pp. 165–168, Dec. 2018, doi: 10.1109/ICEPE.2018.8559776.

[69] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, “Revisiting RCNN: On Awakening the Classification Power of Faster RCNN.” pp. 453–468, 2018.

[70] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[71] D. Mehta *et al.*, “Simple and Efficient Architectures for Semantic Segmentation.” pp. 2628–2636, 2022. Accessed: Jun. 24, 2024. [Online]. Available: https://github.

[72] J. Wang *et al.*, “Deep High-Resolution Representation Learning for Visual Recognition,” *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021, doi: 10.1109/TPAMI.2020.2983686.

[73] R. Padilla, S. L. Netto, and E. A. B. Da Silva, “A Survey on Performance Metrics for Object-Detection Algorithms,” *International Conference on Systems, Signals, and Image Processing*, vol. 2020-July, pp. 237–242, Jul. 2020, doi: 10.1109/IWSSIP48289.2020.9145130.

[74] Jalaj Agrawal, “Mean Average Precision (mAP) Explained in Object Detection,” Medium. Accessed: Jun. 24, 2024. [Online]. Available: https://medium.com/@jalajagr/mean-average-precision-map-explained-in-object-detection-fb61adf67ef4

[75] I. Park and S. Kim, “Performance indicator survey for object detection,” *International Conference on Control, Automation and Systems*, vol. 2020-October, pp. 284–288, Oct. 2020, doi: 10.23919/ICCAS50221.2020.9268228.

[76] O. Corcoll, “Semantic Image Cropping,” Jul. 2021, Accessed: Jun. 24, 2024. [Online]. Available: https://www.researchgate.net/publication/353284602_Semantic_Image_Cropping

[77] Sean McClure, “Building an End-to-End Defect Classifier Application for Printed Circuit Boards,” Towards Data Science. Accessed: Jun. 24, 2024. [Online]. Available: https://towardsdatascience.com/building-an-end-to-end-deep-learning-defect-classifier-application-for-printed-circuit-board-pcb-6361b3a76232

[78] D. Budagam *et al.*, “Instance Segmentation and Teeth Classification in Panoramic X-rays,” Jun. 2024, Accessed: Jun. 24, 2024. [Online]. Available: http://arxiv.org/abs/2406.03747

[79] R. Yang, Z. Pan, X. Jia, L. Zhang, and Y. Deng, “A Novel CNN-Based Detector for Ship Detection Based on Rotatable Bounding Box in SAR Images,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 14, pp. 1938–1958, 2021, doi: 10.1109/JSTARS.2021.3049851.

[80] W. Yao, W. Hua, and H. Wang, “Financial Original Voucher Classification and Verification System Based on Deep Learning,” *Advances in Transdisciplinary Engineering*, vol. 48, pp. 213–221, Feb. 2024, doi: 10.3233/ATDE231331.