MACHINE LEARNING APPROACHES FOR QUALITY CONTROL IN PULP PACKAGING MANUFACTURERS

BY

RATTANA CEESAY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE AND TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2022

MACHINE LEARNING APPROACHES FOR QUALITY CONTROL IN PULP PACKAGING MANUFACTURERS

BY

RATTANA CEESAY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE AND TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2022

THAMMASAT UNIVERSITY
FACULTY OF SCIENCE AND TECHNOLOGY

THESIS

BY

RATTANA CEESAY

ENTITLED

MACHINE LEARNING APPROACHES FOR QUALITY CONTROL IN PULP PACKAGING MANUFACTURERS

was approved as partial fulfillment of the requirements for
the degree of Master of Science (Computer Science)

on June 19, 2023

Chairman	(Nuchjarin Intalar, Ph.D.)
Member and Advisor	(Assistant Professor Thapana Boonchoo, Ph.D.)
Member and Co-Advisor	(Assistant Professor Prapaporn Rattanatamrong, Ph.D.)
Member	(Assistant Professor Saowaluk Watanapa, Ph.D.)
Member	(Associate Professor Punpiti Piamsa-Nga, Ph.D.)
Dean	(Associate Professor Supet Jirakajohnkool, Ph.D.)

Thesis Title	MACHINE LEARNING APPROACHES FOR QUALITY CONTROL IN PULP PACKAGING MANUFACTURERS
Author	Rattana Ceesay
Degree	Master of Science (Computer Science)
Department/Faculty/University	Computer Science Faculty of Science and Technology Thammasat University
Thesis Advisor	Assistant Professor Thapana Boonchoo, Ph.D.
Thesis Co-Advisor	Assistant Professor Prapaporn Rattanatamrong, Ph.D.
Academic Year	2022

ABSTRACT

In molded pulp packaging manufacturing, defect detection and classification processes are critical to ensuring the products meet quality criteria. Yet most manufacturers still rely on human-based manual visual defect classification which can be inconsistent and labor intensive. In this research, we introduce the conjunction of machine vision hardware and machine learning to build a conceptual framework for an automated molded pulp packaging defect detection system. The conceptual framework consists of two modules. First, the image acquisition module setups appropriate hardware and configuration such that high-quality images can be acquired. The second is a machine learning module that constructs a deep learning model with hyper-parameter tuning to automatically detect the defects on the surface of molded pulp products. Our proposed model is based on deep learning model - the Xception architecture, which is recently developed and expected to be more robust on defect detection. In comparison with Traditional machine learning algorithms - SVM and Naive bayes have been widely used in the field of industrial detection. The oriented FAST and rotated BRIEF (ORB) and Bag-of-Visual-Word (BoVW) are implemented for pre-feature extraction. Since molded pulp packaging has obstacles on surface fluctuation by color, grain pulp fiber and non-repeating defect pattern, the Negative Monochrome (NGMC) image preprocessing is proposed to enhance the visibility of defects on the surface and reduce undesired features. The extracted features must be able to describe and distinguish images categories, which could be a limitation for traditional algorithms that required pre-feature extraction. The results demonstrate that the Xception model trained with NGMC images resolution 192x192 and learning rate 0.001 achieved more than 92.98\% accuracy and best generalize across datasets from different production lots, which suggests that the robustness of our conceptual framework has the potential to be utilized in industrial applications.

Keywords: Defect detection, Machine vision, Molded pulp packaging, Pulp and paper packaging industry, Machine learning, Deep learning

ACKNOWLEDGEMENT

This research was fully supported by a research grant from the Faculty of Science and Technology at Thammasat University. I would like to express my very great appreciation, as it would not have been possible without the exceptional support from our advisor, Assistant Professor Dr. Thapana Boonchoo and our co-advisor, Assistant Professor Dr. Prapaporn Rattanatamrong with their enthusiasm, knowledge, and attention to our research details that have been an inspiration and kept our work on track. I am also deeply grateful to my committee chairman and members, Dr. Nuchjarin Intalar, Assistant Professor Dr. Saowaluk Watanapa and Associate Professor Dr. Punpiti Piamsa-Nga for their constructive comments and suggestions in this study. The authors would like to thank you for the support and cooperation from Sugarcane Ecoware Co., Ltd. for information and samples of molded pulp packaging. Moreover, I would also like to thank the staff of the Faculty of Science and Technology at Thammasat University for supporting us on all documentation processes. Finally, I wish to thank my family for their support and encouragement throughout my research.

Rattana Ceesay

TABLES OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

The molded pulp packaging market is rising due to the demand for biodegradable and eco-friendly packaging alternatives to plastic packaging [42]. It is made from sugarcane bagasse, a common agricultural by-product in agrarian countries. As it consists only of water and wood fibers, molded pulp is a renewable material and could be a biodegradable solution. In the manufacturing process, bagasse pulp is formed into desired shapes, e.g., bowl, tray, plate, and clamshell. Product quality control by visual inspection is then performed to detect products with possible defects, e.g., oil spots and particle contaminants, as shown in Figure 2.5.

The defect detection process is crucial since major failure from this process could lead to customer complaints and business loss. Nowadays, the visual inspection is still performed manually by well-trained operators who inspect product piece by piece to ensure product quality. Unfortunately, the "human visual inspection" is not effective and unreliable due to multiple factors e.g., human error, varied operators' angles of view, visual fatigue, defects that are not detectable by the human eye, high labor cost and worker shortage. Due to these factors, the visual inspection process for the manufacture of molded pulp packaging is at risk. The quality control is imperative in any industry. For traditional defects detection techniques, various image processing methods can be applied on an image of product. However, there are many drawbacks; e.g., some image processing methods for defect detection give better results but consume more time, which is not acceptable in a real-time environment. For each type of defect, different techniques are applicable to gain an efficient result. No such technique can classify all types of defects at once (Sanghadiya and Mistry, 2015) However, innovative defect-detection technique of machine vision and machine learning algorithms represented by deep learning, have rapidly developed and thrived in recent years for automated defect detection tasks due to their versatility, lower costs, and lack of reliance on human assistance, but they still depend on large amounts of training data for updates and tuning to improve the inspection performance of the model algorithm (Yang, Li, Wang, Dong, Wang, & Tang, 2020).. Therefore, many researchers are still trying to build and tune model parameters that provide robust defect detection applications for specific industries. For example, in fabric industry where defect detection was done by human with limited rate up to 12 meters per minute, repetitive job and wasted human resources thus a new detection method YOLOv4 model algorithm is proposed (Liu, Wang, Li, Gao, and Li, 2022) or in steel industry where Faster R-CNN model algorithm was proposed to solve current problems of steel surface defect detection which was slow and low precision (Zhao, Chen, Huang, Li, & Cheng, 2021). An automatic visual inspection based on CNN was also proposed by Park et al. (Park, Kwon, Park, & Kang, 2016) for generic approach to dirt, scratches, burrs, and wears detection on steel part surfaces in production line.

Unlike most cases, e.g., in the fabric and steel industries, where the detected surface is smooth, stable in color, and repeated in defect patterns, the challenge of molded pulp packaging is not only fluctuating grain surface and unstable product color due to variation of raw material lots but also non-repeated defect size and patterns as shown in Figure 2.5 and Figure 2.6. To solve major problem of defect detection in pulp packaging industry which process is mostly rely on human, we aim to perform an end-to-end implementation of machine vision and machine learning algorithm to develop conceptual framework for automated molded pulp packaging defect detection system in a way that may potentially serve as an industry-specific practical case study.

1.2 Objective and Scope

In this research, our goal is to introduce an end-to-end implementation of machine vision hardware and machine learning algorithm to build a conceptual framework for automated molded pulp packaging defect detection system in the way that potentially serve as an industry-specific practical case study.

1.2.1 Objective of the Research

The objects of the research are as follows:

1.2.1.1 To introduce the integration of machine learning and machine vision algorithm to build a conceptual framework for automated defective product classification in molded pulp packaging manufacturing industry.

1.2.2.2 Design project's hardware setup include machine vision selection criteria, camera configuration and LEDs light sources setup comparison to understand the effect of light on perceptibility and visibility of defect on product surface.

1.2.2.3 Exploration and compare performance of deep learning - Xception model on image dataset which collected from different machine vision camera configuration; RGB and Negative Monochrome.

1.2.2.4 Analysis the performance of deep Learning algorithm; Xception architecture with hyper-parameters tuning in comparison to Supervised Learning; SVM and Naive Bayes using ORB and Bag-of-Visual-Word (BoVW) approach on defective molded pulp packaging classification task.

1.2.2 Scope of the Research

The scopes of the research are as follows:

1.2.2.1 This research is limited to only a particular model of Sugarcane Bagasse molded pulp packaging, named "32 oz burrito bowl" (Fig 6).

1.2.2.2 The product dataset is randomly collected from a real production line for a period of three months or more.

1.2.2.3 The research algorithm is based on Deep Learning; Xception with hyper-parameters tuning in comparison to Supervised Learning; SVM and Naive Bayes with ORB and Bag-of-Visual-Word (BoVW) approach.

1.2.2.4 The research has its limitations in that it only uses an experimental hardware setup and does not aim to develop a field defect classification platform that can be readily used in a real production scenario.

1.3 Significance of Research

The author sincerely hopes that this study will be beneficial and useful, that it can help both the molded pulp packaging manufacturer and researcher to have case study for benchmarks and gain a better understanding of how they can improve molded pulp packaging's industry productivity, reduce the cost and save time by implementation of machine vision and machine learning algorithm.

CHAPTER 2
REVIEW OF LITERATURE

2.1 Problem of the study

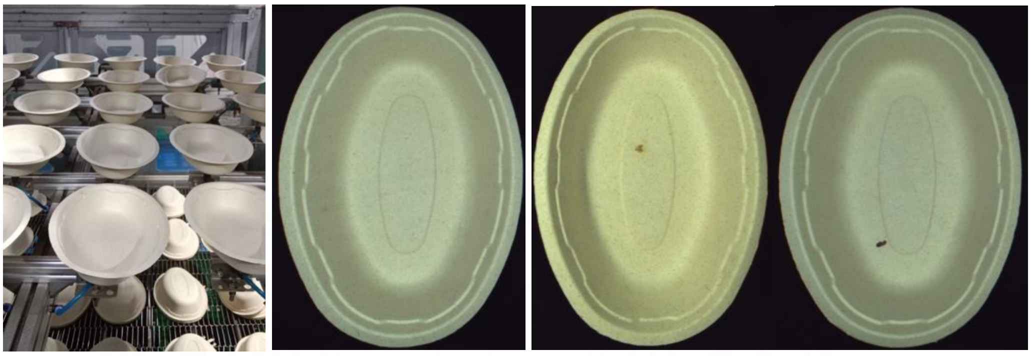
In the molded pulp packaging industry, visual inspection for quality control is a critical step. It allows defects to be detected early in production, reducing the risk of costly product recalls. On the other hand, failure of this process could result in customer complaints and business losses. Currently, classifying products that have anomalies and flaws is done by human inspectors, as shown in Figure 2.1. Examples of defective products are shown in Figure 2.1

Figure 2.1

Conventional visual inspection process of molded pulp packaging industry

Figure 2.2

Molded pulp packaging product output and its categories (a) Products output from manufacturing process (b) Non-defective product (c) Defective product e.g., oil spots and particle contaminant



(a) (b) (c)

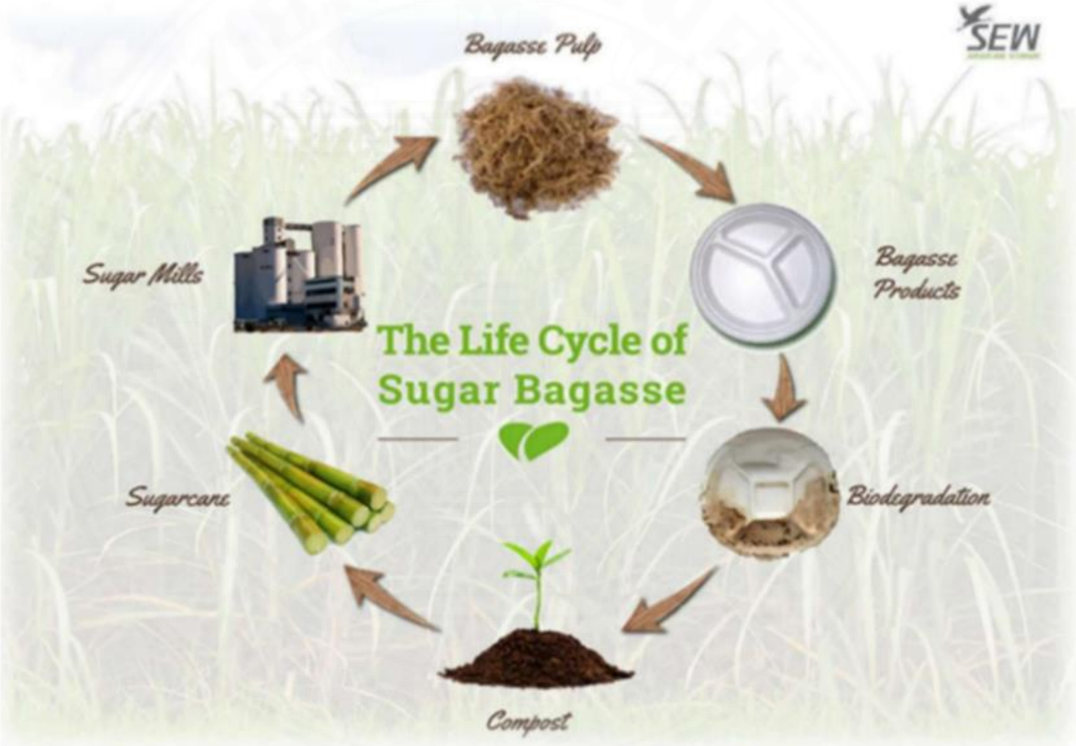
The industry faces a number of issues as a result of the outdated, labor-intensive, and slow conventional process, including the possibility of human error, the inability for workers to be available 24/7, and the high cost of labor. Additionally, since agricultural raw materials are readily available in third-world countries, molded pulp packaging is often produced there and exported to first-world nations. Major defects in molded pulp packaging products have the potential to harm customer confidence, reduce customer satisfaction, decrease business bargaining power, and result in a financial loss due to product returns and sales decline. Hence, detecting defects is a core competency that a company should possess in order to improve the quality of its manufactured products. Research on an automatic defect-detection system has obvious benefits over conventional manual detection; it can reduce production costs, improve production efficiency, and improve product quality, as well as build a solid foundation for the intelligent transformation of the manufacturing industry.

2.2 Molded Pulp Packaging Manufacturing Overview

The source of raw materials for the production of molded pulp packaging is non-wood pulp which is on rapid growth due to global concerns about deforestation including agricultural waste and non-woody plant materials e.g., grass, straws, reeds and sugarcane bagasse. Sugarcane bagasse which is the fiber left over after the juice has been extracted from the sugarcane plant as shown in Figure 2.3, is a common agricultural by-product in Thailand and other agrarian countries. Sugarcane bagasse gains high preference as a raw material for molded pulp packaging because it is inexpensive and inexhaustible for the production of pulp due to lower silicon content therefore easier to make pulp and uses less chemicals when compared to other types of fiber materials.

Figure 2.3

The Life Cycle of Sugarcane Bagasse pulp and paper industry.

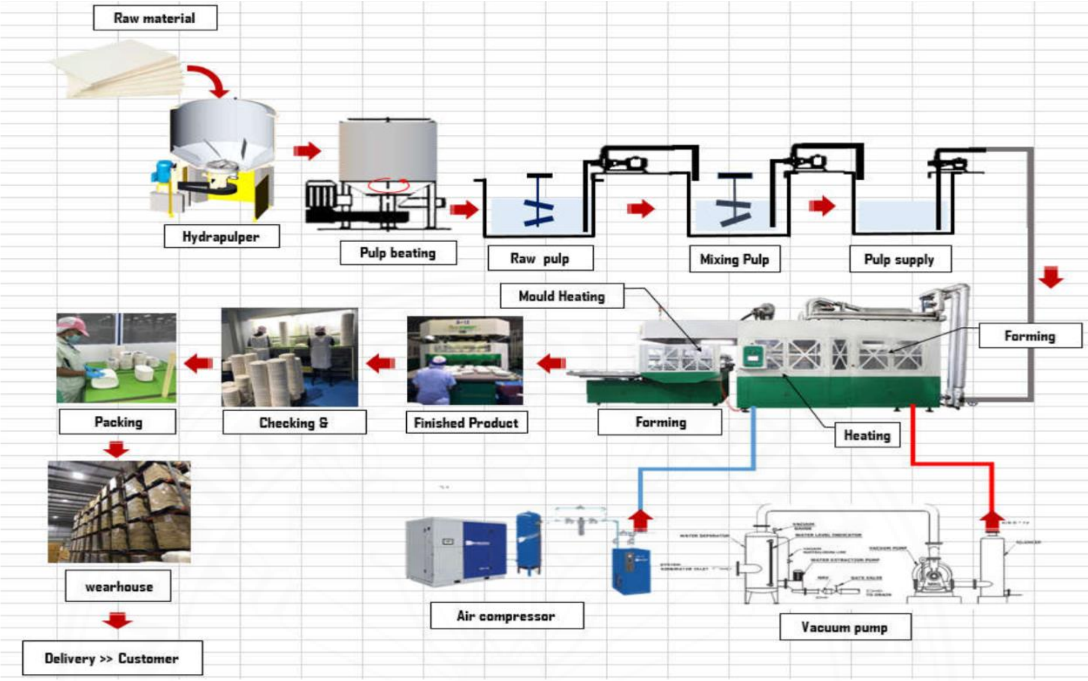


Source: How To Make Paper From Sugarcane Bagasse. Data from Pulp. (2018). (http://www.paper pulping.com/news/How-to-make-pulp-with-bagasse.html)

In the manufacturing process, which is depicted in Fig.4, raw materials are mixed, products are formed into desired shapes, and heated molds are used to press the pulp for a smoother finish and better dimensions. The finished product is going through a visual inspection for quality control before packing and delivery to the customer.

Figure 2.4

Typical molded pulp packaging product manufacturing process



2.2.1 Defective Products in Molded Pulp Packaging

Surface defects on molded pulp packaging products result in rejects. Some rejects could be reworked, while others must be scrapped. In both cases, it will cost the company. Rejects from production lines mostly come from the following defects:

2.2.1.1 Oil Spot — deposits of oil that have fallen onto the product from some processing step and are typically round or an irregular shape.

2.2.1.2 Particle contaminant / Dirt impurities — most of the small dark brown impurities and black specks caused by foreign substances that have nothing to do with the raw material itself. (Figure 2.5 & 2.6)

Figure 2.5

Example of defects size between 0.8 – 3 mm (a) Oil Spot defects. (b) Particle contaminant / Dirt impurities.

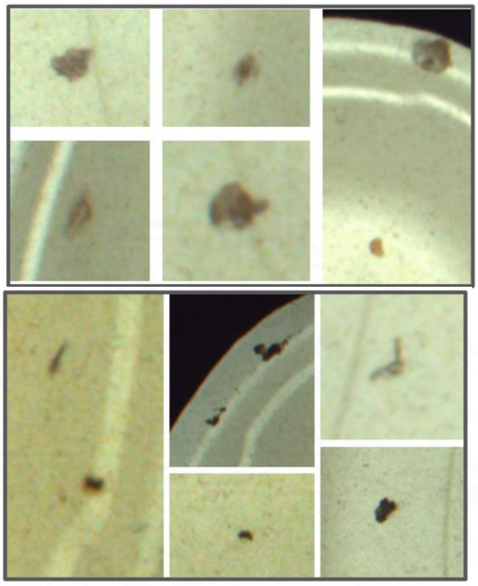


Figure 2.6

Example of Sugarcane Bagasse molded pulp packaging, named "32 oz burrito bowl" Estimate surface area 58,600 mm^2

2.3 Machine Learning Based Visual Inspection

An automated visual inspection system based on machine learning consists of two modules. The image acquisition module converts the target objects placed in the light field into images and transfers them to a computer, and the machine learning module applies machine learning algorithms to build a defect detection model. The image acquisition module consists of the two following components:

2.3.1 Machine vision camera is the first key technologies for visual inspection and defect detection since obtaining high quality images is essential for detection system's effectiveness in order to emphasize an important feature of the objects and diminish undesired ones. Machine vision camera detection technology can improve the detection efficiency, enhance real-time performance and accuracy of detection.

2.3.2 Light-source selection is one of the key important to enhance the visibility of certain features and reduce undesired one especially imbalanced shade hence researcher must consider the interaction between light and objects. Currently, LED (Light Emitting Diode) light source have become available for most of machine vision application with various shapes and by far the most widely used lighting types in machine vision. LED technology has improved in stability, intensity, and cost-effectiveness compare to others common light-source (Illumination, 2023).

2.4 Choosing a Machine Vision Camera

One of the key technologies for visual inspection and defect detection is the machine vision camera, which automatically collects images of real objects using optical devices and noncontact sensors. Machine vision cameras, particularly for large-scale industrial applications, can improve detection efficiency, enhance real-time performance and accuracy of detection, and reduce labor requirements. Since obtaining high-quality images is essential to the automated visual inspection system's effectiveness, it is important to emphasize important features of the objects and diminish undesirable ones. Choosing the right camera is the most crucial task. To choose the right camera, several factors must be taken into account, including pricing, working environment (e.g., working distance & working area), and camera option (e.g., sensor, type, size) that suit working applications.

2.4.1 Camera Working Environment

To choose a suitable Machine Vision Camera, first the user must be able to define their working distance – distance between camera and object, and field of view (FOV) - the working area that the target object is placed on referring to the x and y-axis. Then the user can do the following calculation.

2.4.1.1 Megapixel (MP) Calculation and Resolution Selection

Since higher megapixel cameras can capture images that show fine changes in higher contrast and provide higher quality of image to be used for inspection and classification while the camera price is also increased with higher megapixel. Thus Megapixel (MP) must be calculated to find the suitable one.

The pixel resolution means how many millimeters each pixel, expressed by the following equation [6]:

$$\text{Pixel resolution} = \frac{\text{Size of field of view in the Y direction (mm)}}{\text{Sensor pixel count in the Y direction}}$$

For example: If FOV working area equal to horizontal (x) 200mm & vertical (y) 300mm, the vertical working field of view (y) 300mm will be used for calculation. In case user have the image sensor type 2594 x 1944 (5 MP), the vertical sensor (y) = 1944 pixels will be used for calculation.

$$\text{Pixel resolution} = \frac{300\text{ mm}}{1944\text{ pixels}} = 0.15\text{ mm /pixel}$$

The minimum detectable size on a camera sensor (CCD or CMOS) is 1 pixel. For defect detection applications, it is recommended to have a detection capability of at least 2-4 pixels per defect (Keyence corporation, n.d.) in order for machine learning algorithms, especially those deep learning that take each image pixel as input, to be able to learn and distinguish whether a product is defective or not.

$$\text{Detection capability}=\frac{0.15 \text{ mm}}{1 \text{ pixels}}\times 4= 0.6 \text{ mm Area}$$

Based on the calculations, if the inspection requires detection of defect that are as small as 0.6 mm with a field of view (y) 300 mm a camera with a resolution of 5MP or more is needed.

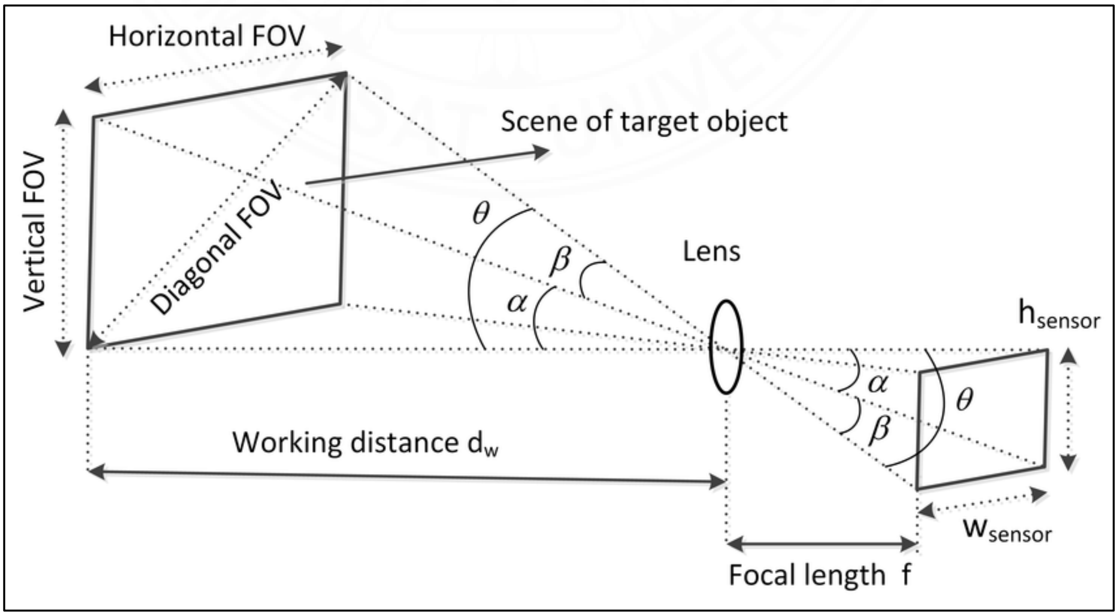
2.4.1.2 Focal Length Calculation and Lens Selection

The size of the field of view (FOV) is the area captured on an inspection target [6] which can be changed by the lens used. The lens must be selected to suite with working application, working distance and camera resolution. Once user calculated suitable camera resolution, the focal length must be calculated to choose the right lens.

To calculate focal length user must provide input included: Working Distance (dw), Sensor size / Image size (Camera Specific) and Object Size (Vertical FOV and Horizontal FOV) in order to choose lens with suitable focal length (f) shown in Figure 2.7.

Figure 2.7

Relation between Field-of-view, working distance, focal length of lens and camera sensor size.



Source: Ngo, Abdukhakimov, & Kim, (2019)

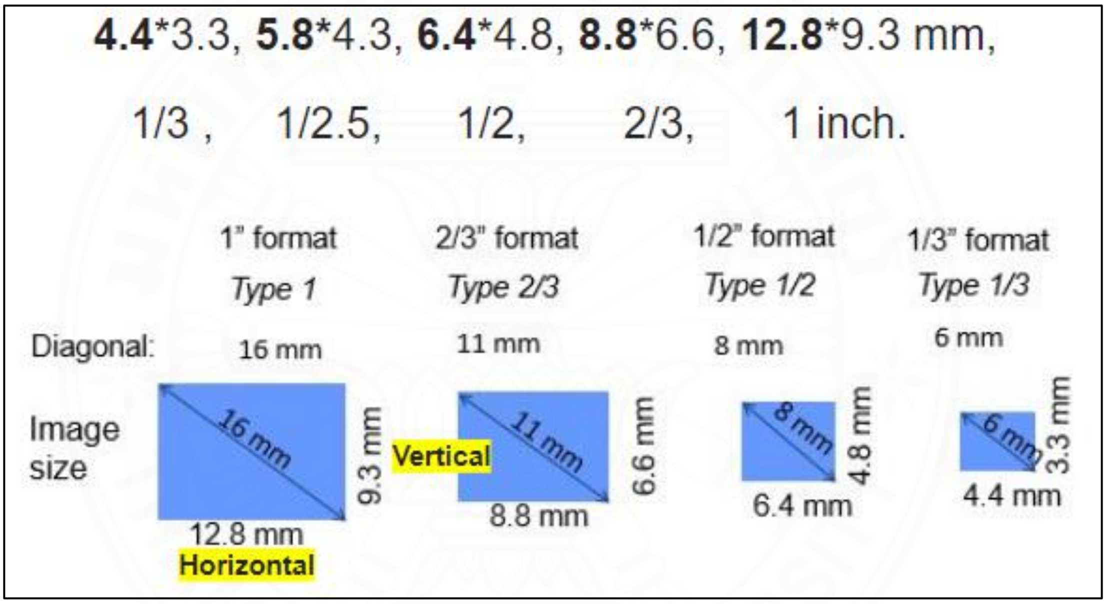
1) Working Distance (dw) or Object distance is measured from the lens's front principal plane to the object itself.

2) Object Size is real-size of an object, given in mm which is define as Horizontal FOV and Vertical FOV.

3) Typical Sensor size / Image size which is Camera Specific as shown in Figure 2.8.

Figure 2.8

Typical Sensor size / Image size of camera.



Source: Chouinard (2023)

The calculation must perform with following criteria

1) Calculate only one axis horizontal or vertical which is related to angle of view

2) Object distance and Image size is constant value

An Angle of view result will be different depend on calculation axis; horizontal or vertical but Focal Length result will be the same. The Focal Length calculation formula as shown in Figure 2.9.

Figure 2.9

Focal Length calculation formula.

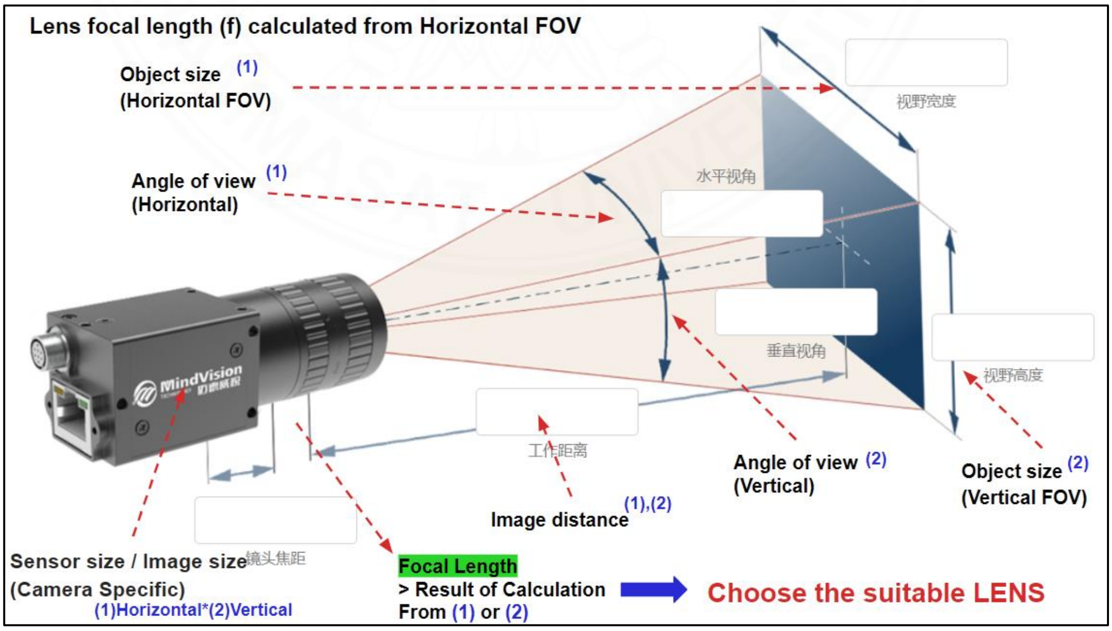
1/Focal length = 1/Image distance + 1/Object distance	**Image distance**, **Object distance** (mm)	Magnification = Image size / Object size = -(Image distance / Object Distance)	**Object size** (mm) **Image size** (mm)
Focal length = (Object distance / ((1 /Magnification) + 1)) * 1000	**Object distance** (mm) **Magnification** (No Unit)	Angle of view = (180/π) * 2 * aTan(Image size / (2 * Focal length * (Magnification + 1)))	**aTan(x) -** arc tangent

Source: Zaborowska (2023)

The relation between field-of-view – horizontal, vertical and angle-of-view view – horizontal, vertical as shown in Fig.10. The calculation must be done on the same axis to get right result. Once user got the Focal Length, the suitable lens can be selected.

Figure 2.10

Relation between field-of-view, angle-of-view and focal length of Machine Vision Camera.



Source: Shenzhen MINDVISION technology co., ltd (n.d.).
https://www.mindvision.com.cn/jtxx/list_108.aspx?lcid=18&lcids=2414]

2.4.2 Machine vision camera option requirement

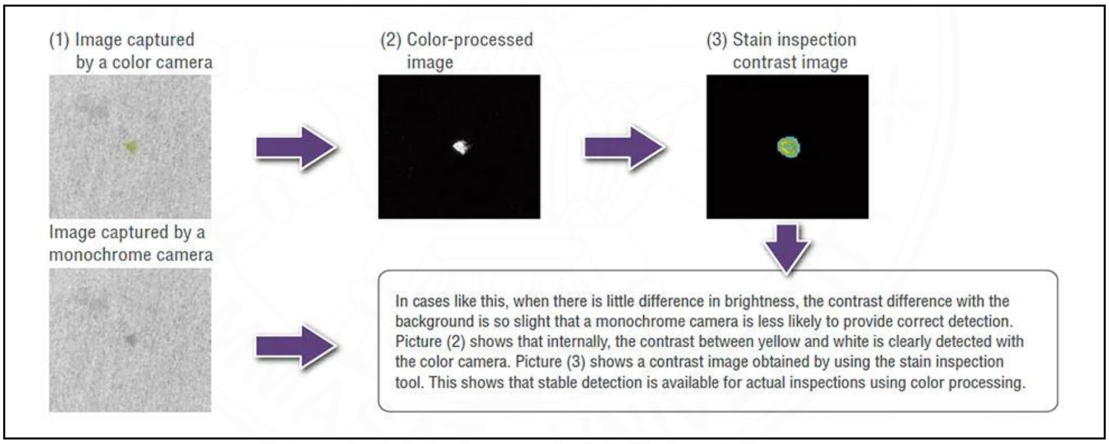
Every machine vision camera application is different, so the user must consider the camera option that meets their requirements. It is important to understand a few key camera characteristics and features that can help the user recognize the factors needed to take into consideration.

2.4.2.1 Selecting based on color or monochrome type image sensor

Selecting a camera type is whether to use a color or monochrome type. If the differences at the sensing points are detected based on hue, the color cameras may have an advantage. Example in Fig.11. of using color processing to detect a yellow stain on a white base, which is not easily detectable by a monochrome camera.

Figure 2.11

Using color processing to detect a yellow stain on a white base. [6]



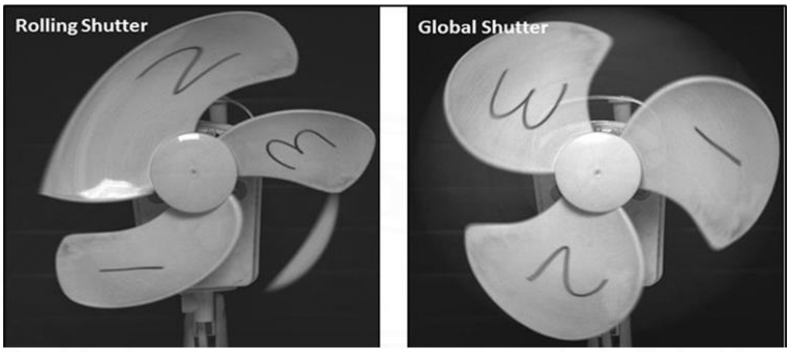
Source: Selecting the Correct Camera | Machine Vision Basics | KEYENCE America. (n.d.). https://www.keyence.com/ss/products/vision/visionbasics/tips/primer1/

2.4.2.2 Selecting Global Shutter vs. Rolling Shutter

Rolling shutter exposes the pixel rows in a certain order while a global shutter exposes each pixel to incoming light at the exact same time. So biggest advantage of the global shutter over the rolling shutter, it does not suffer from the same distortion effects as shown in Figure 2.12, it's easier to sync with peripheral devices, because there is a single point in time when exposure starts. Downside of a global shutter is more costly and originally only available on the more expensive CCD sensors whereas CMOS sensors used rolling shutters.

Figure 2.12

Distortion effects between Rolling Shutter and Global Shutter.



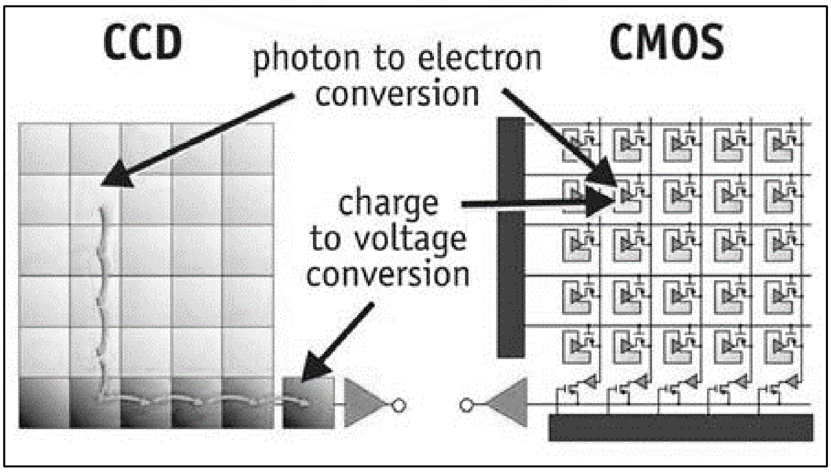
Source: Vandendorpe (2022)

2.4.2.3 Selecting CCD Sensor vs. CMOS Sensor

The two main types of electronic image sensors are the charge-coupled device (CCD) and the active-pixel sensor (CMOS). Difference lies in the way each pixel value is read. For a CCD sensor, pixel values can only be read on a per-row basis. Each row of pixels is shifted, one by one, into a readout register. Conversely, for a CMOS sensor each pixel can be read out individually as shown in Figure 2.13.

Figure 2.13

Read process different between CCD and CMOS Sensor.



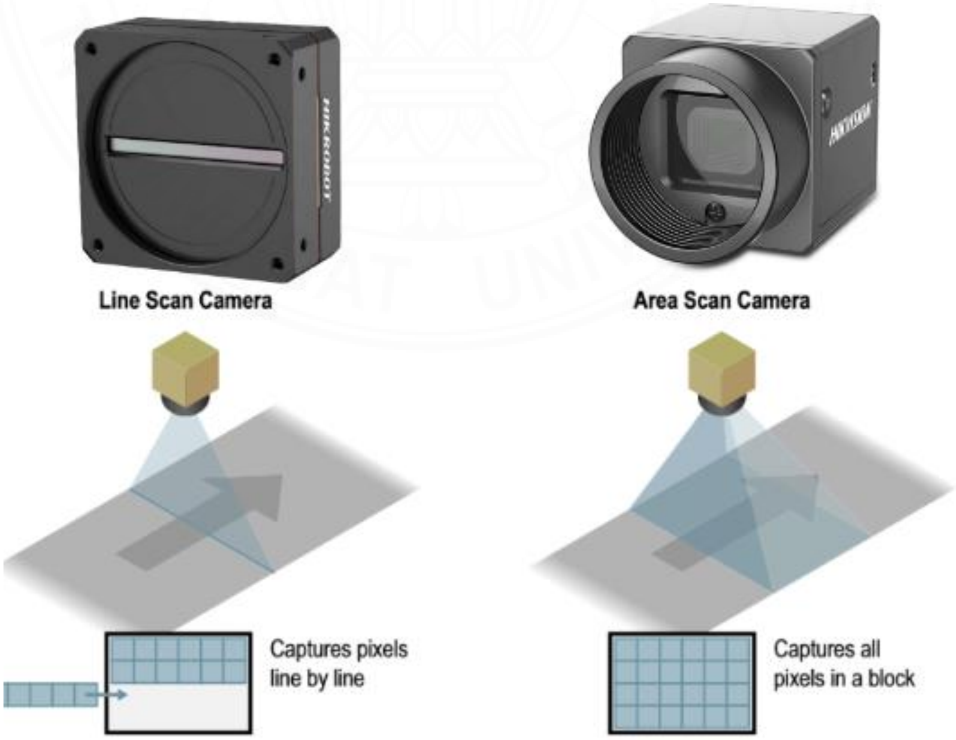
Source: Shakeri, Ariannejad, Sedaghati, & Amin, (2012)

2.4.2.4 Selecting based on camera type & size

Compact are reduced in size, but are equipped with the same specifications as larger-sized cameras. Compact types are primarily selected to efficiently use in limited installation space. In cases where machine vision is to be installed in the available space of an existing facility. It is beneficial to use compact cameras to fit in a limited area without changing the machinery. Area scan and Line scan camera as shown in Fig.14. Area scan cameras, a rectangular-shaped sensor captures an image in a single frame. The resulting image has a width and height that directly corresponds to the number of pixels on the sensor. Because of this, area-scan cameras are suited for machine vision applications, where the objects are small and have almost the same size in both dimensions. While line scan cameras contain a single row of pixels and build the final image pixel line by pixel line with specific advantages, inspecting round or cylindrical parts which may require multiple area scan cameras to cover the entire part surface.

Figure 2.14

Working procedure of Area scan and Line scan camera.



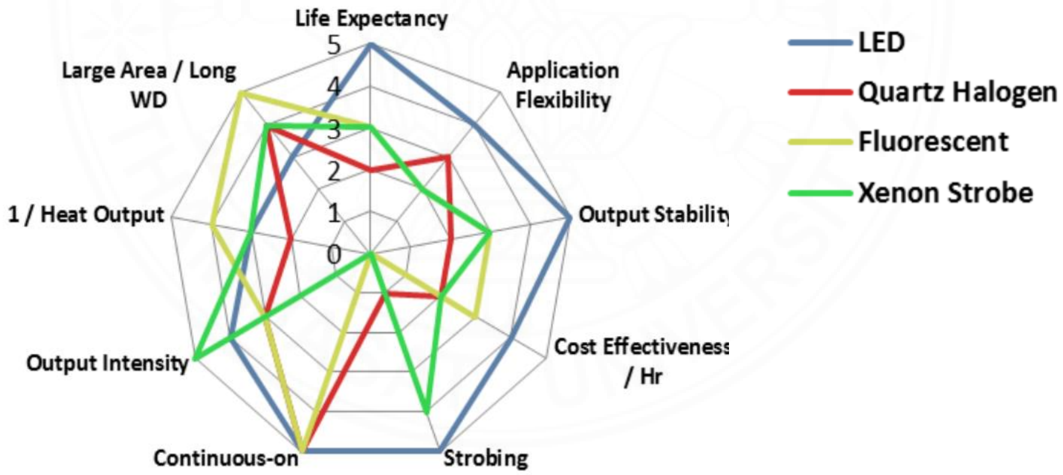
Source: Vandendorpe (2022)

2.5 Machine Vision Light Source

A visual inspection system is based on an image, and the key to success lies in getting high-quality images with high visibility of certain features. Thus, it is important to consider the interaction between light and objects. The quality of lighting is a critical factor in creating a quality, robust, and timely vision inspection. Currently, LED (light-emitting diode) light source devices have become available for most machine vision applications in various shapes and are by far the most widely used lighting types in machine vision, particularly for small to medium-scale inspection stations. LED technology has improved in stability, intensity, and cost-effectiveness compared to other common light sources (Illumination, 2023), such as xenon, fluorescent, and quartz halogen, as shown in Figure 2.15.

Figure 2.15

Comparison and contrast of common vision lighting sources.



Source: Illumination, A. (2023) A Practical Guide to Machine Vision Lighting - Advanced Illumination. https://www.advancedillumination.com/a-practical-guide-to-machine-vision-lighting/

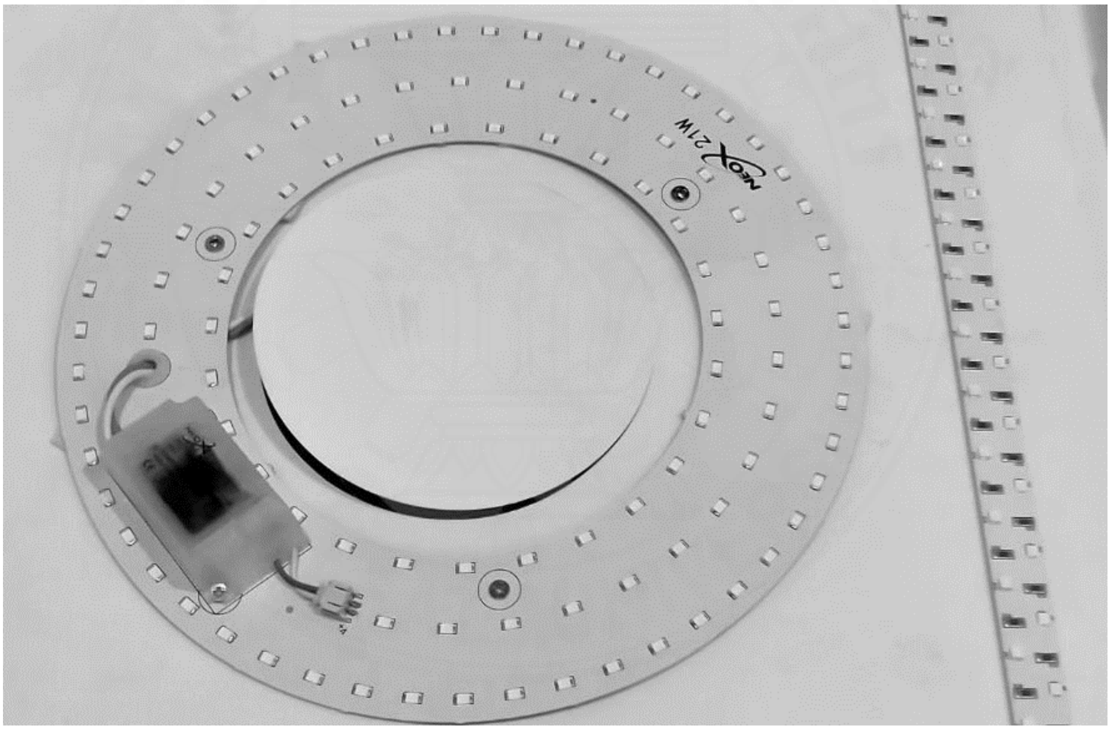
An LED light source can be customized to be suitable with project applications. The popular light source is a circular ring array of LEDs and linear array of LEDs (Ren, Fang, Yan, & Wu, 2021) as shown in Figure 2.16.

1) The circular ring array of LEDs - It provides high brightness, can be conveniently installed, and can effectively avoid shadow occurrence and enhance the features to be detected. It was used in the following application: For example, IC chip appearance detection and printed circuit board (PCB) substrate detection.

2) The linear array of LEDs - It is commonly used, has good heat dissipation and flexibility of usage. It can be used for defect detection of some large structural parts. For example; copper strip and steel sheet.

Figure 2.16

Example of LEDs used in our research (a) Circular ring array of LEDs (b) Linear array of LEDs



The contrast of the image and target features can be enhanced by effectively selecting the best one or combining light sources. A proper understanding of the application and its requirements is necessary to choose the light source. A correct selection may significantly affect the development time, cost, efficiency, and quality of images produced.

2.6 Defect Detection in Manufacturing Overview

Industrial product defect detection is an important part of manufacturing. Due to unavoidable casualty factors in the manufacturing process or other human factors, lead to the industrial defects such as surface defects, scratches, and spots influence the reliability of the industrial or use, causing serious and harm to life. Therefore, it is particularly important to carry out industrial product defect detection.

2.6.1 Typical Defect Detection Techniques

Typical defect detection technologies often used in traditional manufacturing industry include direct defect detection and pattern recognition-based defect detection (Gong, Bai, Liu, & Mu, 2020).

1) Direct defect detection technology mainly includes X-ray detection, ultrasonic detection and magnetic particle detection. It is a technology that can directly detect and determine whether a product has defects

2) Defect detection technique based on pattern recognition is mainly automatic optical inspection method, by choosing appropriate light source and industrial camera collection and the surface of the product image, through some image recognition and processing algorithms to extract the work piece defect feature information, compared with zero defect work pieces.

The most common typical defect detection used in industrial include:

1) Automatic Optical Detection

Automatic optical inspection technology is based on the human eye vision imaging and discriminant principle of the human brain intelligence by optical illumination of measured object and image sensor technology to obtain information, through digital image processing to enhance the target features, and then adopt the method of pattern recognition, by using some image processing algorithms extract characteristic information from the background image, and are classified and characterized, and then feedback to perform control mechanism, to realize the classification of the product, group or separation, the quality control in the process of production, etc. Automatic optical surface defect inspection technology has been widely applied in industrial, agricultural, biological, medical and other industries, especially in precision manufacturing and assembly industry, such as LCD, silicon wafers and, PCB.

2) X-ray detection

X-ray detection is a nondestructive detection method that use the different attenuation characteristics of the matrix material and the defects of the work piece to find the defects. After more than 100 years of development, X-ray detection technology has formed a relatively complete X-ray detection technology system consisting of X-ray photography, X-ray real-time imaging, digital plate X-ray imaging, and X-ray computer tomography, etc. from the original film radiography technology. X-ray detection has no strict requirements on the surface finish of the work piece, and the material grain size has little influence on the detection results. It can be applied to the detection of internal defects of various materials, so it has been widely used in the welding quality inspection of pressure vessels.

Although the typical defect detection method can accurately detect some of the known defects. But they have drawbacks on poor generality and limit to specific defect categories. It cannot be directly applicable to different kinds of defect. When come to some new defects or problems, it needs to manually design new test program.

2.6.2 Machine Learning Based Defect Detection

In contrast to the drawbacks of typical defect detection methods, machine learning, especially deep learning, can be more easily adapted to different kinds of defects and on a larger industrial product dataset to improve testing efficiency, reduce labor costs, and promote the integration of information. The algorithms that could be used for image classification and defect detection tasks in an industrial include:

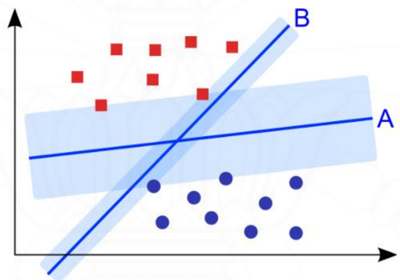
2.6.2.1 Supervised Machine Learning algorithm

(1) Support Vector Machines (SVM)

Support Vector Machines (Luckert, 2016) belong to the area of supervised learning methods and therefore need labeled, known data to classify new unseen data. The basic approach to classify the data, starts by trying to create a function that splits the data points into the corresponding labels with (a) the least possible number of errors or (b) with the largest possible margin. This is due to the fact that larger empty areas next to the splitting function result in fewer errors, because the labels are better distinguished from one another. The datasets may very well be separable by multiple functions without any errors. Therefore, the margin around a separating function is being used as an additional parameter to evaluate the quality of the separation. In Figure 2.17, separation function A is the better one, since it distinguishes the two classes in a more precise manner. Formally, Support Vector Machines create one or multiple hyperplanes in an n-dimensional space. The first attempt in the process of splitting the data is always, to try to linearly separate the data into the corresponding labels. If the data is completely separable in linear fashion, the resulting function can be used to classify future events.

Figure 2.17

Visualization of a Support Vector Machine splitting a data set into two classes, by using two different linear separations.



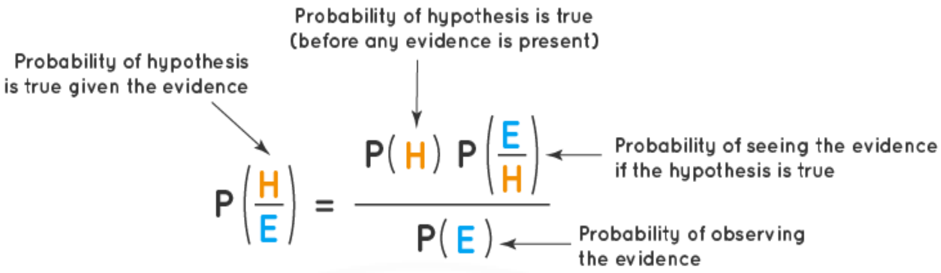
Source: Wikiwand - Support Vector Machine. (n.d.). https://www.wikiwand.com/de/Support_Vector_Machine

(2) Bayesian Networks (Naive Bayes)

Bayesian networks (Luckert, 2016) are probabilistic directed acyclic graphical models. The models consist of nodes and directed connections between these nodes that symbolize dependencies between them. Each node represents an attribute of interest for the given task. Normal Bayesian networks use known data to estimate the dependencies between attributes and the class label and use this information to calculate probabilities of possible different outcomes of future events. It automatically applies the Bayes' theorem as shown in Figure 2.18 to complex problems and is therefore able to gain knowledge about the state of attributes and their dependencies.

Figure 2.18

Bayes Theorem probability formulation



$$P\left(\frac{H}{E}\right) = \frac{P(H)\,P\left(\frac{E}{H}\right)}{P(E)}$$

Source: Bayes Theorem - Formula, Statement, Proof | Bayes Rule. (n.d.). Cuemath. https://www.cuemath.com/data/bayes-theorem/

2.6.2.2 Bag-of-Visual-Word (BoVW)

Feature extraction from images is required for supervised learning algorithm based for image classification. The principle of feature extraction is to obtain the most relevant features from the image data to obtain a sufficient and robust descriptor. Feature extraction is a crucial technique in the field of computer vision and image processing tasks like image classification. Feature extraction is a challenging subject as the features vary significantly due to several factors like noise, variations and scale (Kabbai, Abdellaoui, & Douik, 2018). The bag of visual words (BoVW) model has proven to be efficient for image classification since it can effectively represent distinctive image features in vector space (Sultani, & Dhannoon, 2021). The image has keypoints or local features identified as prominent image regions that have rich local information (such as color or texture), and these features can be detected using different detection and description method. In our research, BoVW using Oriented Fast and Rotated BRIEF (ORB) descriptors are adapted for image classification. ORB, an efficient alternative to SIFT or SURF, is a fusion of the FAST keypoint detector and the BRIEF descriptor with many modifications to enhance its performance. The approach has three main steps:

1) Extracting keypoints from original images by using the OpenCV ORB (Oriented Fast and Rotated Brief) feature detection and description algorithm to create a descriptor for each extracted keypoints. Then making clusters from

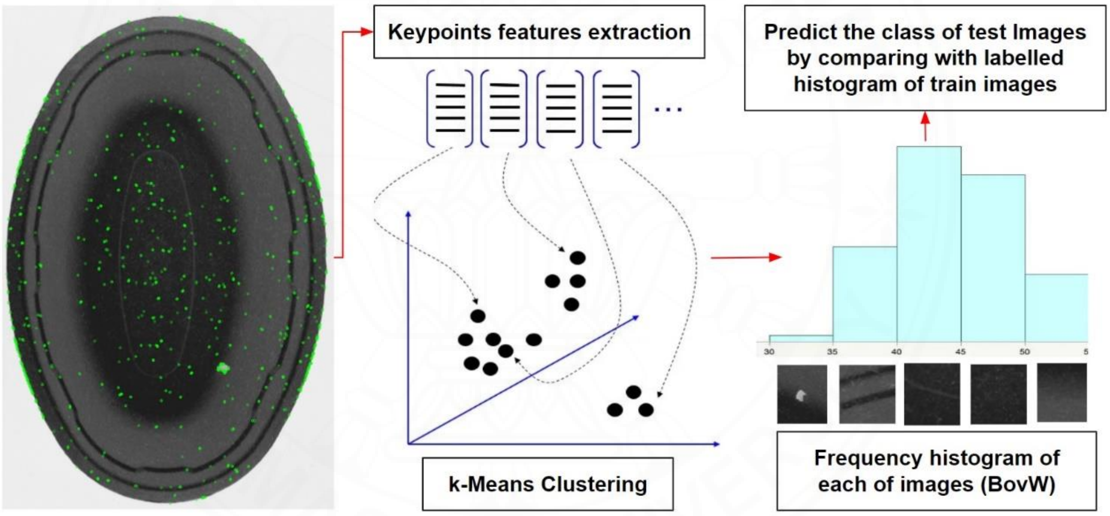
keypoint descriptors of training images by using the unsupervised k-means clustering algorithm. The center of each cluster is considered as the visual vocabularies dictionary.

2) Generating visual words frequency histogram from the vocabularies and their frequencies in the image to represent each labeled image in the training set as shown in Figure 2.19.

3) Then the frequency histograms are used to train our SVM & Naive Bayes models. Finally, the models are used to classify images from the test set by comparing the extracted frequency histogram of unlabeled images with labeled frequency histogram from the training stage.

Figure 2.19

Molded pulp packaging features extraction process based on BoVW



2.6.2.3 Deep Learning algorithm

(1) Convolutional Neural Network (CNN)

Basic CNN architecture, model consists of an input layer, convolutional layers, RELU layers, pooling layers and fully-connected layers which connected in simple sequence. The following are definitions of each layer in architecture shown in Fig.20

1. INPUT Layer: this layer is responsible for the input of raw pixel values (image width, height, with three color channels R, G, B or one greyscale channel) of the image.

2. CONV layer: the input image is placed into a set of convolution filters, each of which activates certain features in the image.

3. RELU layer achieves faster, more efficient training by mapping negative values to zero and maintaining positive values. Sometimes this is called activation because only activated features can be transferred to the next layer.

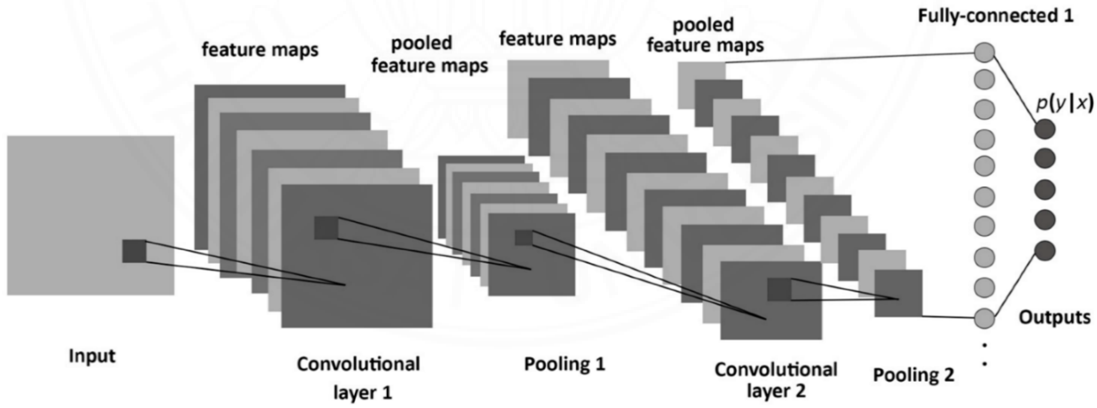
4. POOL layer: down-sampling on spatial dimensions (width and height) to reduce memory usage.

5. Fully-Connected layer: the fully connected layer is to flatten the previous results and then receive the most basic neural network

Compared to supervised learning algorithm such as SVM and Naïve bayes, the CNN obtains much better results and can work properly with different types of defects which made deep convolutional neural network is widely used in defect detection and has excellent effect on the identification of defective parts in industrial products.

Figure 2.20

The diagram of a Basic CNN architecture



Source: Albelwi & Mahmood (2017)

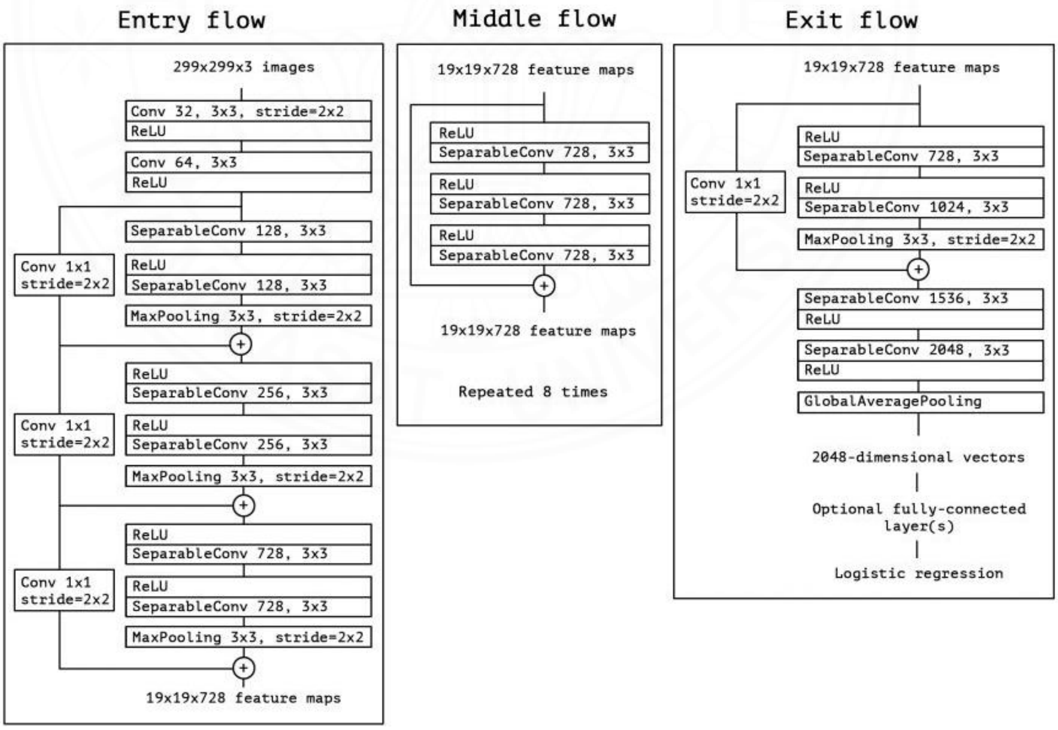
(2) Xception deep convolutional neural network

The Xception model was proposed by Google (Chollet, 2017), Inc. and was implemented to solve defect classification problem in this paper. It has a depth of 126, including 36 convolutional layers to extract features. A global average pooling layer is used to replace the fully-connected layer to reduce the number of parameters, and the softmax function is used to output the prediction. The 36 convolutional layers are structured into 14 modules, all of which have linear skip connections around them, except for the first and last modules. The 36 convolutional layers are divided into 3 components: entry flow; middle flow and exit flow. The data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. The entry flow consists of 8 convolutional layers, the middle flow consists of 8*3= 24 convolutional layers and the exit flow consists of 4 convolutional layers as shown in Figure 2.21.

The Xception model applied depthwise separable convolution and shortcuts between Convolution blocks as in ResNet, which can significantly reduce the convolution operation cost and has overperformed VGG-16, ResNet and Inception V3 in most classical classification challenges.

Figure 2.21

The architecture of Xception network.



Source: Ioffe & Szegedy (2015)

2.6.2.4 Evaluation of Machine Learning

Once models building is finished. One of the important parts in machine learning, is the problem of how to indicate which of model results were appropriate for its application. The following will show different types of evaluation value available. The calculation formula of each evaluation value and structure of confusion matrix are shown in Figure 2.22.

(1) Accuracy and misclassification rate

Describes the relative amount of truly and falsely classified data in a dataset.

(2) Precision values

Also called positive prediction value, is defined as the relative amount of correctly as true classified instances among all as true classified instances.

(3) Recall values

Also called sensitivity is defined as the relative amount of as true classified instances among all true instances.

(4) F-Measure (F1-score)

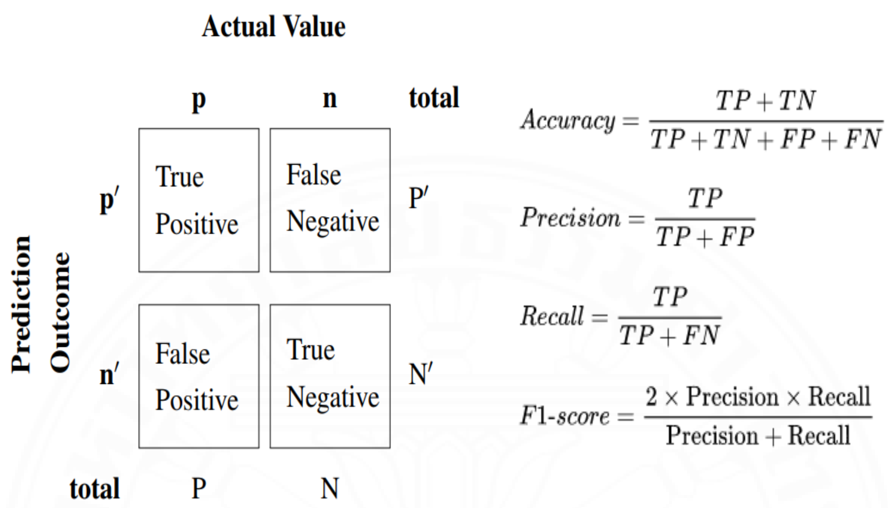
Aims to combine the statements of recall and precision by using the harmonic mean between the two.

(5) Confusion matrix

The best approaches to illustrate the performance of machine learning programs is the confusion matrix, also called contingency table, which distinguishes between true positive, false positive, true negative and false negative predictions.

Figure 2.22

(a) Structure of confusion matrix (b) calculation formula of evaluation value: Accuracy, Precision, Recall and F-Measure.



Source: CIS520 Machine Learning | Lectures / Precision Recall. (n.d.). https://alliance.seas.upenn.edu/~cis520/dynamic/2017/wiki/index.php?n=Lectures.PrecisionRecall

2.6.2.5 Hyper-parameters tuning in deep learning

(1) Image Resolution Tuning

The CNN models take images as input, learns visual information of the images, and creates feature maps which are the input for the following layers. Image quality factors e.g., resolution, noise, contrast, blur, and compression, affect the visual information contained in the images. The details preserved in the visual information (e.g., fine defect, the structure of fiber on surface) can vary drastically with the reduction of image resolution (Thambawita, Strümke, Hicks, Halvorsen, Parasa, & Riegler, 2021b). While additional memory requirements make processing of high-resolution images difficult, the scale of features and level of model accuracy also changes as a function of image resolution. To achieve better model performance with lower input image resolutions might initially seem paradoxical, but, in various machine learning paradigms, a reduced number of inputs or features is desirable as a means of lowering the number of parameters that must be optimized, which in turn diminishes the risk of model overfitting (Sabottke, & Spieler, 2020). Thus, finding optimal resolution that suited the model's application is essential to reduce processing time and resource requirements. In general, the resolutions for training CNNs usually range between 32×32 and 512×512.

(2) Learning Rate (LR) Tuning

Learning rate is one of the most important hyper-parameters that controls how much to change the model in response to the estimated error each time the model weights are updated by optimizers algorithms. A weight updated too small may result in a long training process that could lead to the model not learning, whereas a weight updated too large may result in the model converge too quickly to a suboptimal solution or an unstable training process. Therefore, we must find the optimal learning rate.

(3) Batch Size Tuning

Batch Size is an essential hyper-parameter in deep learning. This parameter represents a number of training samples that will be used during the training in order to make one update to the network parameters. Different chosen batch sizes affect both the training-testing accuracies and different runtimes. Choosing an optimal batch size is crucial when training a neural network (Lin, 2022). Using Full Batch Learning in a database with a small sample size is feasible and effective. Nevertheless, once it is an extensive database, feeding all the data into the network at once will cause an explosion of memory. In most cases, it may depend on the features of the data that we will train the model on. Too low or high batch size would be the prime suspect in fluctuations and generalization of the model, because the accuracy would depend on what examples the model learns at each batch. Example in research Masters, & Luschi, (2018) on the CIFAR-10, CIFAR-100, and ImageNet datasets show that increasing the batch size will gradually decrease the range of learning rates, providing steady convergence and acceptable test performance. On the contrary, a smaller batch size provides more gradient computation, yielding more stable and reliable training results. To find an optimal batch size is important for CNN model training process.

(4) Optimizers for convolutional neural networks

The goal of deep learning is to reduce the difference between the predicted value and the actual value which is known as loss function. The optimizers are algorithms used to change the attributes of neural network such as weights and learning rate in order to reduce the losses. To minimize the loss function, we must find the optimized value for weights. There is various deep learning optimizer algorithm e.g., SGD, RMSprop Adam, Adadelta, Adamax and Nadam. In our research we adopt Nadam (Nesterov-accelerated Adaptive Moment Estimation) to acquire advantages from a fusion of Adam optimizer and Nesterov Momentum algorithms which can improves the speed of convergence and requires least exhaustive parameter tuning which favors our focus on optimizer learning-rate (LR) tuning.

1. Nadam optimizer (Brownlee, 2021) is an extension to the Adaptive Movement Estimation (Adam) optimization algorithm to add Nesterov's Accelerated Gradient (NAG) or Nesterov momentum, which is an improved type of momentum and more broadly, the Nadam algorithm is an extension to the Gradient Descent optimization algorithm. The update formula of Nadam optimizer is shown in Figure 2.23. The learning rate (η) is an important hyper-parameter as it is inverse to how much model update the weights to minimize the loss function.

Figure 2.23

The Nadam optimizer's update rule

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\left(\beta_1 \hat{m}_t + \frac{(1-\beta_1)g_t}{1-\beta_1^t}\right)$$

Source: Optimization Algorithms - A Brief Overview:- · GSoC'18 @ CERN. (2018, May 16). https://www.sravikiran.com/GSOC18//2018/05/16/optimizers/

(5) Image Augmentation Methods

Image augmentation is the process to increase the size of the training data using images that already exist in the training set to prevent CNN training problem include (1) lack of a good amount of data is that the deep learning model might not learn the patterns or the functions from the data and hence it might not perform well or (2) the scenario that at some point, the training error will keep decreasing, while the error on the test/hold-out data increases ,which is called overfitting, because the model keeps learning some patterns from the training data that do not generalize on the test data. The basic geometric augmentations that usually used for machine learning model listed below (Shorten & Khoshgoftaar, 2019) and as shown in Figure 2.24.

1. Flipping: horizontal axis flipping is much more common than flipping the vertical axis. This augmentation is one of the easiest to implement and has proven useful on datasets.

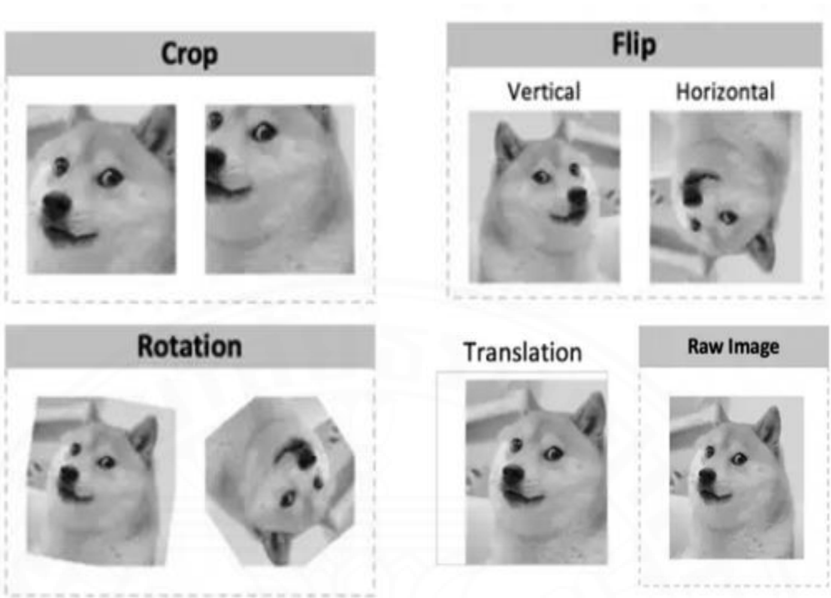
2. Rotation: rotation augmentations are done by rotating the image right or left on an axis between 1° and 359°. The safety of rotation augmentations is heavily determined by the rotation degree parameter. Slight rotations such as between 1 and 20 or −1 to −20 could be useful on recognition.

3. Translation: shifting images left, right, up, or down can be a very useful transformation to avoid positional bias in the data. As the original image is translated in a direction, the remaining space can be filled with either a constant value such as 0 s or 255 s, or it can be filled with random or Gaussian noise. This padding preserves the spatial dimensions of the image post-augmentation.

4. Cropping: cropping images can be used as a practical processing step for image data with mixed height and width dimensions by cropping a central patch of each image. The contrast between random cropping and translations is that cropping will reduce the size of the input such as (256,256) → (224, 224), whereas translations preserve the spatial dimensions of the image. Depending on the reduction size chosen for cropping, it might not be a label-preserving transformation.

Figure 2.24

Basic Image Augmentations.



Source: Chernytska (2023)

2.6.2.6 Stratified KFold Cross Validation

The k-fold cross validation creates the process where every sample in the data will be included in the train-test set at some steps (Figure 2.25). We do not want to use a specific dataset to fit the models due to the particular dataset fitting better on one model design. Instead, we want to use multiple datasets to fit, resulting in multiple weighted models of the same design. The weighted model with best generalization at satisfied accuracy will be used in real-world scenario.

Figure 2.25

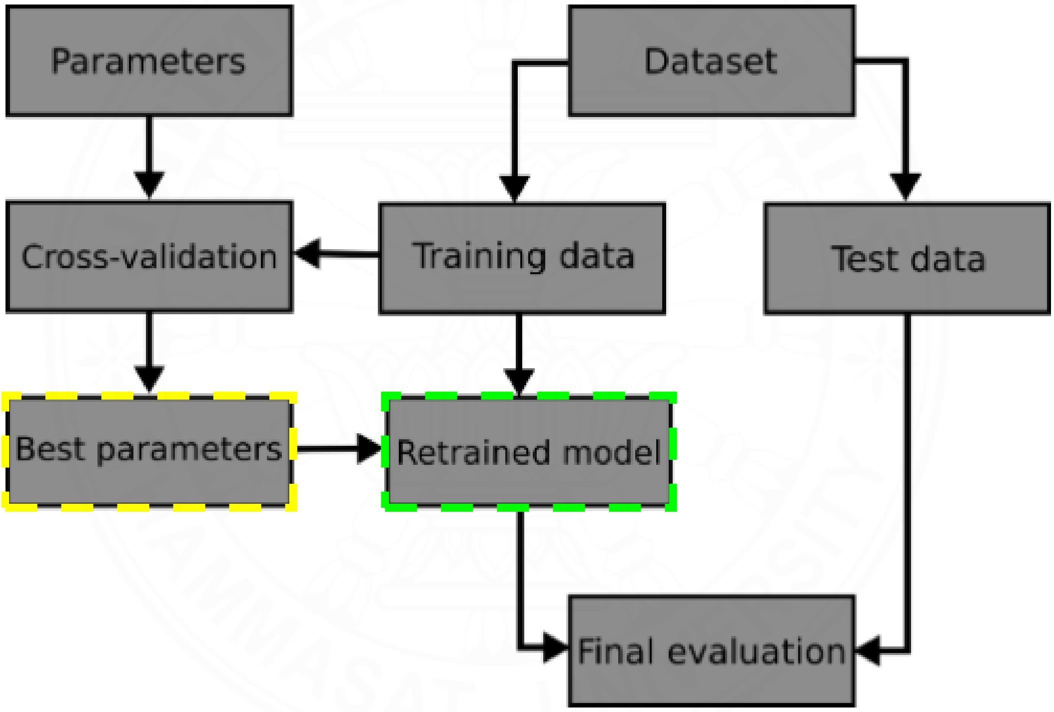
Example on splitting the training dataset into k subsets in 5-fold validation

Source: Data adapted from Pandian, S. (2022). K-Fold Cross Validation Technique and its Essentials. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/02 /k-fold-cross- validation-technique-and-its-essentials

The stratified k-fold is improved from standard k-fold validation. The stratified intended to solve the problem of imbalanced target classes. The algorithm attempts to balance the number of instances of each class in each fold. During the training stage, the performance is printed for each model. Then model and weight are stored. The best weighted model's parameter will be retrained for final evaluation (Figure 2.26).

Figure 2.26

Machine learning pipeline for stratified k-fold cross-validation



Source: Data adapted from Cross-validation: evaluating estimator performance. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/cross_validation.html

However, cross validation is often not used for evaluating deep learning models because of the greater computational expense. For example, k-fold cross validation is often used with 5 or 10 folds. As such, 5 or 10 models must be constructed and evaluated, significantly adding to the evaluation time of a model. Hence in our research we will implement stratified k-fold during the final stage of training after hyper-parameter tuning completed.

2.7 Autonomous Defect Detection Algorithm

In recent years, the improvement of computing power and the advent of big data, machine learning has been successfully integrated with machine vision on image classification tasks to solve manufacturing industries problem included productivity optimization, fault diagnosis and especially in our case of defective product detection and quality control. Nowadays, a growing number of researchers are focusing on machine learning approaches on defect detection related problems (Cadavid, Lamouri, Grabot, Pellerin, & Fortin, (2020). In early years, classical machine vision approaches using supervised machine learning such as SVM, KNN and Naive Bayes are commonly used classifiers for industrial purpose. Currently and further development, deep learning represented by convolutional neural networks (CNN) models such as ImageNet, VGGNet, GoogLeNet and ResNet have played an important role in defect detection. The CNN was originally designed for image analysis is now most popular architecture hence a good fit for automated defective product image classification in industrial (Ren, Fang, Yan, & Wu, 2021). Many methods to classify defective product in manufacturing industries have been proposed such the author of Burresi, Lorusso, Graziani, Comacchio, Trotta, & Rizzo, (2021) was aims to solve problem in embedded devices production. A transfer learning approach was adopted to detect missing screws on devices using the CNN YOLOv3 and TINY-YOLO models. The training image went through an augmentation process to simulate variations in lighting conditions, affine transformations (translation, rotation, scaling, shear mapping), hue/saturation changes and occlusions. The results show that YOLOv3 and TINY-YOLO models achieved the same 97.0% accuracy. And another study was based on CNN ResNet50 (Dang, Wang, Lee, & Wang, 2021) for defect detection on ski goggle lens surfaces. Five cameras with Spherical light were used to capture all regions on the surface. The high-resolution captured image of size 4094 x 3000 pixels has been cropped as the region of interest (ROI) for labeling data before going through the CNN Machine Learning Module. Result shown that the higher the image resolution, the better the performance and data augmentation also helps improve performance. The result 94.34% accuracy rate was highest which came from CNN's ResNet50 model trained on dataset of 256 x 256 pixels.

Over the years, there has been a trend where the deeper the model is, the better performance the model can get. In 2014, at the ImageNet competition, the VGG16 model with a depth of 23 resulted in a top-1 classification error of 28.5%. In 2015, the Residual Network (ResNet) model with depth 168 resulted in a top-1 classification error of 24.1%. In 2016, the Inception V3 model with depth 159 resulted in a top-1 classification error of 21.8%. And finally, in 2017, the Xception model with 126 layers resulted in a top-1 classification error of 21.0% (Team, n.d.). As author of Lo, Yang, & Wang (2019) mentioned that the Xception model is a recently developed special CNN architecture that is more powerful with less overfitting problems than the current popular CNN models. However only a few use cases of the Xception model can be found in literature. In 2021, the methodology was proposed to train neural networks Xception and Unet for defect detection and classification of steel workpieces (Boikov, Payor, Savelev, & Kolesnikov, 2021). The artificial datasets of steel slab defects were generated to train the models. Then the models were tested on real data and showed good results with 0.81 accuracy. In some research deep learning model is also being compared with traditional machine learning model like in research Arshad, Obaid, Gull, & Shahzad, (2022). which proposed improved machine learning methodologies for Steel Defect Classification using Machine Learning (SDCML) to devise improved methods that classify whether the image belongs to a defect category or not. The supervised machine learning algorithm KNN was compared with deep learning algorithm CNN VGG-16 feedforward classifier. The results show that among all the algorithms, the accuracy obtained via VGG-16 that achieved 97.54% accuracy which is better than KNN that obtained 41.11% accuracy.

There are still limited studies in defect detection on objects where the surfaces have fluctuating grains and do not repeat in patterns. The research (Prasitmeeboon & Yau, 2019), was conducted on defect detection of particle boards whose surfaces shown grain and texture of wood in order to improve human visual inspection which was not effective. The particle sample boards were imaged inside a light-box to ensure uniform diffuse lighting and cropped to 3200 x 4000 pixels. The classifier must avoid false negatives, avoiding misclassifying defective as a defect free one. Result shown SVM tends to misclassify only one class but the KNN classifier misclassified both at same rate.

From the previous research of machine learning and machine vision on defective product detection, the problem in molded pulp packaging is unique and has never been implemented since. Thus, our research's main contributions are to introduce an end-to-end implementation of machine vision hardware and machine learning algorithms to build a conceptual framework for an automated defect detection system in the quality control process of the molded packaging manufacturing industry, which is currently a human-manual process. First, we setup the hardware platform, including machine vision camera selection, camera configuration, and LED light source setup. The negative monochrome (NGMC) camera configuration is applied to prevent color problems in products that change across raw material lots and to generalize the model learning pattern. Then we build and analyze a deep learning model using the Xception architecture with hyper-parameter tuning, including image resolution, learning rate, and batch size tuning, to find the optimal classification model performance. The effects of each hyper-parameter on the Xception model's performance are reported and discussed. In comparison to deep learning, we build defect classification models using supervised learning algorithms, SVM, and Naive Bayes. The ORB (Oriented FAST and Rotated BRIEF) and Bag-of-Visual-Word (BoVW) techniques are implemented for feature extraction. Finally, we analyze the model's performance and provide our suggestions on its implementation in a real manufacturing scenario.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Research Workflow

In our research to build a conceptual framework for automated molded pulp packaging defect detection systems, we focused on classifying images of defective and non-defective molded pulp packaging product named "32oz burrito bowl" as shown in Fig.6. by introducing the conjunction of machine vision hardware and machine learning. The conceptual framework consists of two modules. First "Image Acquisition Module" in Section 3.2, we set the criteria for selecting machine vision and finding the proper hardware platform configurations in order to acquire suitable image datasets. Both camera and light-source configurations must be analyzed to understand their impacts on defect characteristics on the surface of molded pulp material so that the signatures of multiple defects can be learned by the classification model. Then we collect image datasets from two different camera configurations: RGB and Negative Monochrome (NGMC) for classification model performance comparison. Second "Machine Learning Module" in Section 3.4, we will train machine learning algorithms on acquired datasets to build defect detection models. Our research algorithm is based on a deep learning model - the Xception architecture with hyper-parameter tuning. In comparison with supervised learning algorithms - SVM and Naive bayes. The oriented FAST and rotated BRIEF (ORB), and Bag-of-Visual-Word (BoVW) are implemented for pre-feature extraction for supervised learning models. For Xception models, the following hyper-parameter tuning techniques are applied: image resolution adjustment, learning rate tuning, batch-size tuning, image augmentation, and optimizer alternation. After that, the accuracy and precision of each model are evaluated and compared against each other. Finally, we provide the conclusions of this research, our perspective on how to utilize the model in real industry practice, and suggestions for further research.

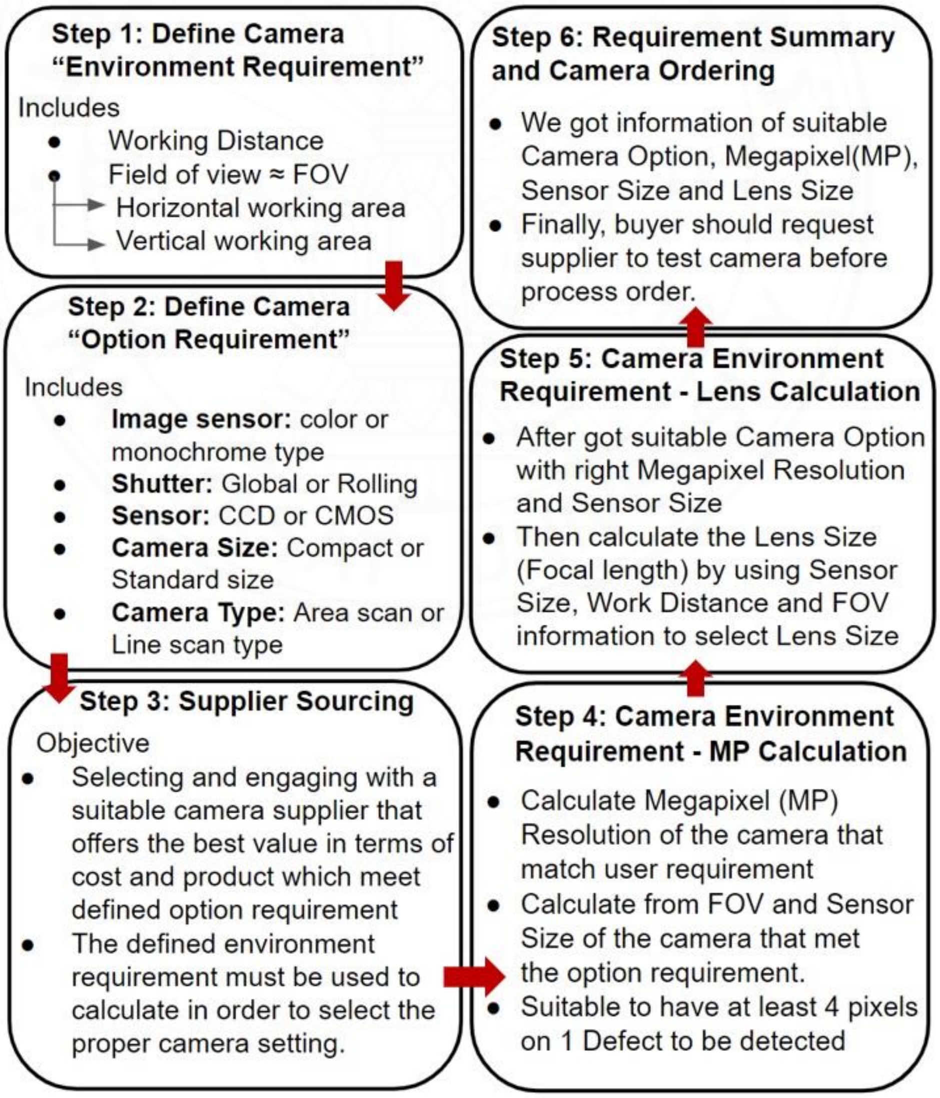
3.2 Image Acquisition Module

3.2.1 Machine Vision Hardware Platform

To choose the suitable project's machine vision, several factors must be taken into account, including pricing, working environment, e.g., working distance & working area, and camera option, e.g., sensor, type, size that suit the working applications. Our proposed machine vision hardware selection process to prepare a suitable camera is described in detail as the flowchart in Figure 3.1. The detail of each step was explained in Chapter 2 Section 2.4 "Choosing a Machine Vision Camera".

Figure 3.1

Research proposed machine vision hardware selection process



In our research, machine vision was selected following research's proposed selection process. Our machine vision camera was selected to be suitable for the project's smallest inspection at 0.8 mm following our research's proposed selection process shown in Figure 3.2 and Figure 3.3. And our experimental hardware setup, as shown in Figure 3.4.

Figure 3.2

Project Selected Machine Vision Camera based on proposed selection process

Equipment	Specification
Camera: MV-GE502C Type: Compact type - Area Scan Brand: MindVision	Resolution: 2592*1944 (5 Mega Pixels) Sensor size: 1/2.5" Sensor type: CMOS Shtter type: Rolling Shutter
Machine Vision Lens: MV-LD-4-4M-G Brand: MindVision	Focal length: 4mm Target size: 1/1.8" Minimum object distance: 0.3m Aperture: F=1:2.0 ~C

Figure 3.3

Hardware detail of project selected machine vision camera

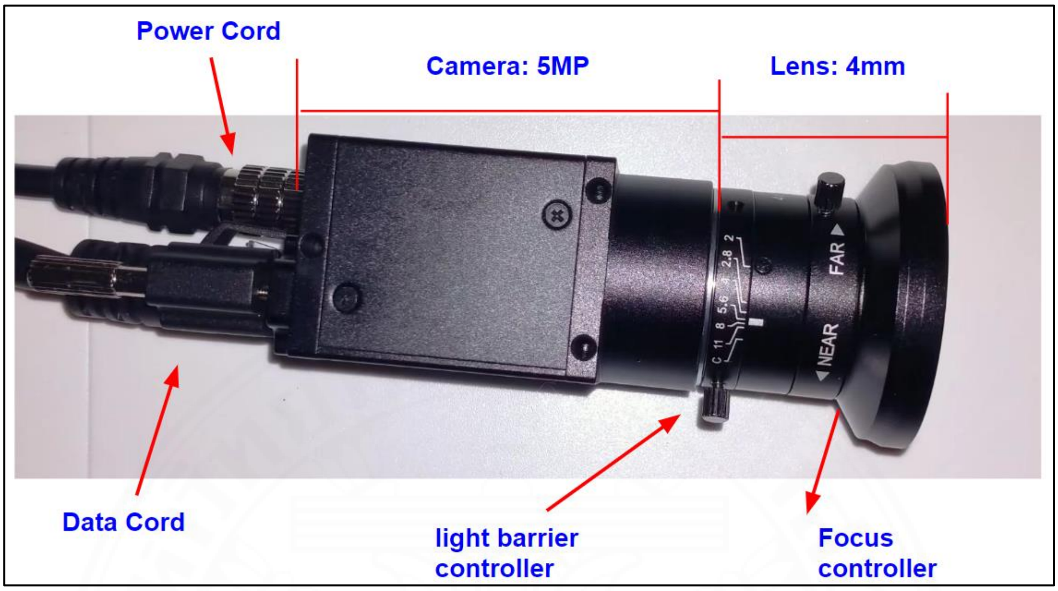
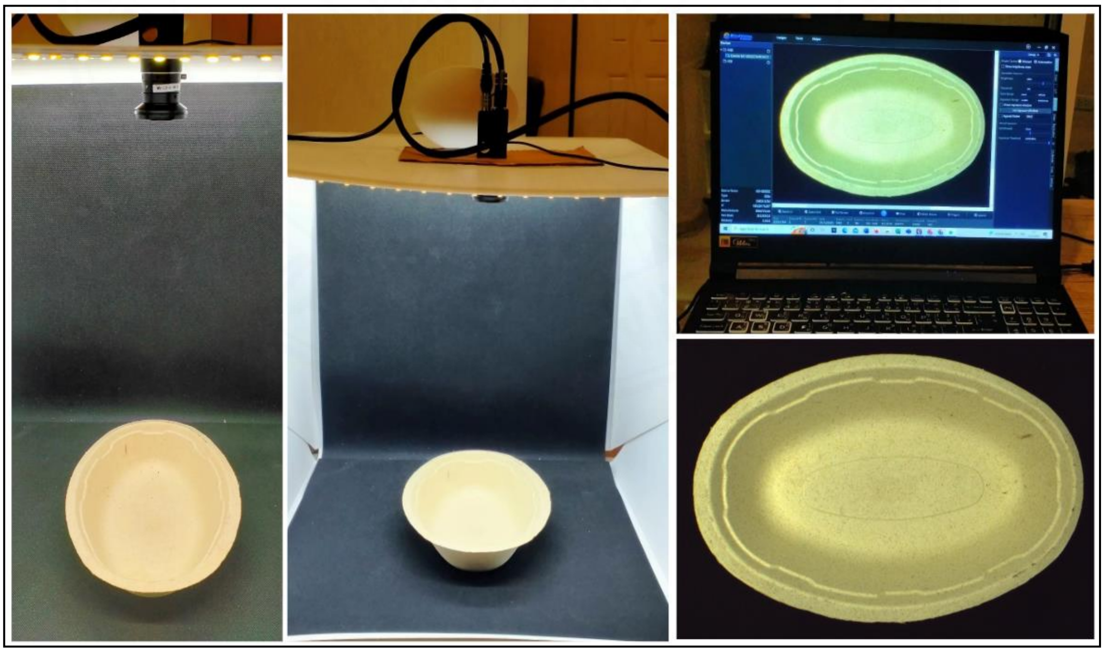


Figure 3.4

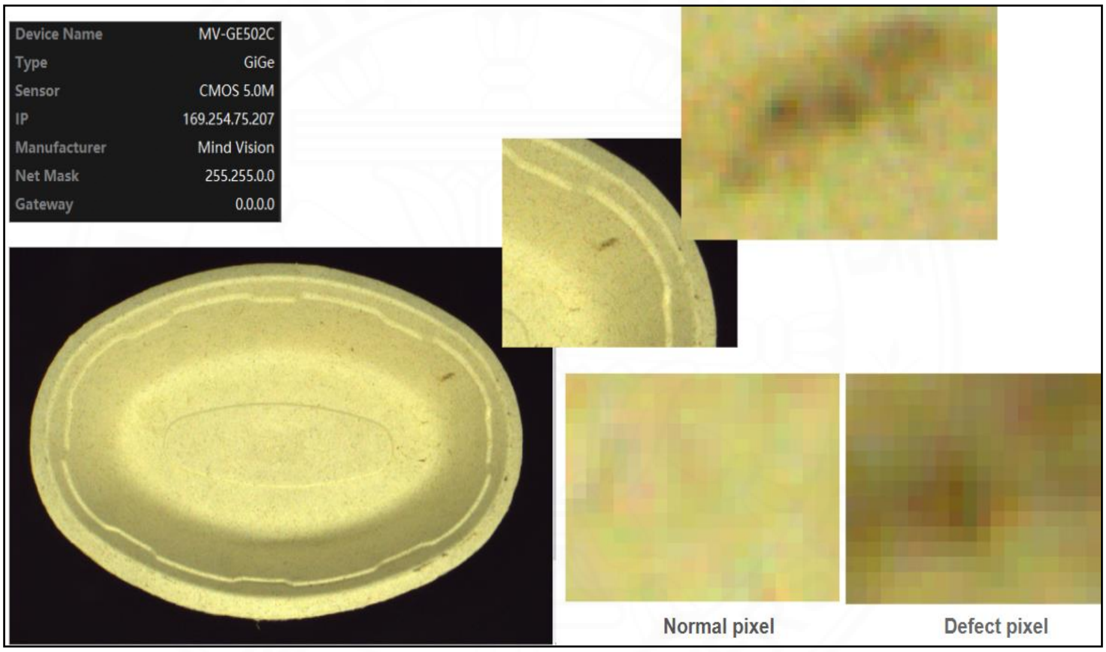
Project Experimental Hardware Setup



The images of defective products captured by the project machine vision camera, as shown in Fig.31 could verify the performance of the camera, which met the selection criteria that there must be at least 4 pixels on the defect target to differentiate defective and non-defective pixels. We also found that the surface of the product is curving, which causes shade on the sides, different in color, and varies in surface texture. Especially for products from different production lots, we can obviously find surface fluctuations by shade, pulp fiber distribution, and non-repeating defect patterns, as shown in Figure 3.5.

Figure 3.5

Image result of defective molded pulp packaging product under project selected machine vision camera



3.2.2 Light Source Configuration & Surface Analysis

The main objective of the LEDs light-source platform is to make the defect features of the target objects visible and reduce undesirable features, especially imbalanced shade on objects, which could cause issues on the training model. The angle between the light beam and the object surface causes the illumination effect and shadow. We compare different LEDs types - circle-center and line-side on our platform, as shown in Figure 3.6. We will analyze its effect on our object as a means to choose appropriate LEDs types.

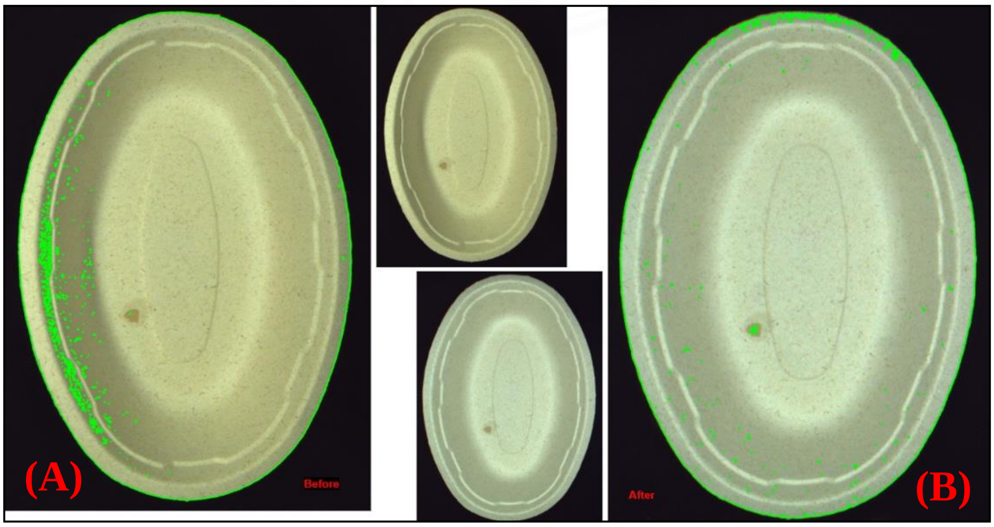
Figure 3.6

Effect of different LEDs source on target object

For an image analysis technique, we implement the image processing tool OpenCV, which is one of the most widely used tools for computer vision and image processing tasks. By implementing Image Thresholding and Contour techniques to analyze light & shade distribution on the object surface, as shown in Figure 3.7.

Figure 3.7

Implement of OpenCV on molded pulp packaging for image and light-source analysis task (A) Linear array of LEDs setup (B) Circular ring array of LEDs



We performed image feature analysis on the visibility of defect features and the distribution of light & shade on the object surface. Along with light-source adjustment between different light-source setups. The circular ring array of LEDs is selected for our hardware platform because it can effectively avoid the shadow phenomenon and provide balanced light & shade distribution, which highlight the features to be detected.

3.2.3 Machine Vision Camera Configuration

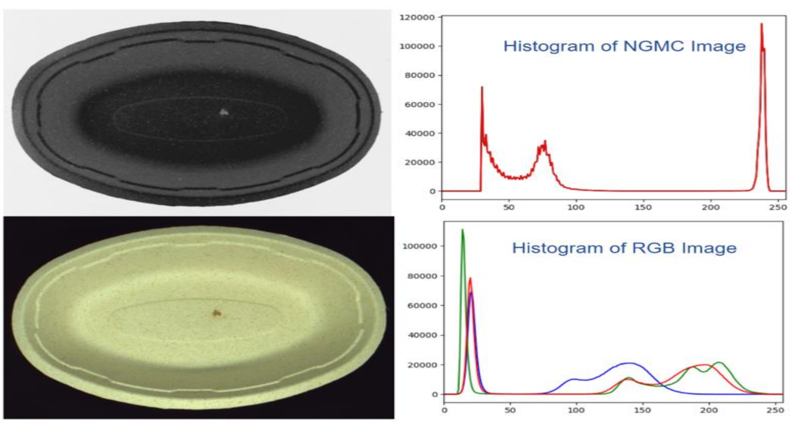
In comparison with RGB, we applied Negative Monochrome (NGMC) camera configurations as shown in Figure 3.8 to prevent color problems in products that change across raw material lots, to emphasize defects, and to generalize the model learning pattern. The configurations have a great advantage for machine learning based molded pulp packaging defect detection, including:

3.2.3.1 Build image datasets that more generalize across different production lots by representing images using one single color instead of different hues to reduce the effect of product color shifting due to raw material color variation.

3.2.3.2 Reduce the learning features of the machine learning model from RGB (3 channels) to Monochrome (1 channel), which reduces the number of inputs that must be optimized by the model for the classification task. (Figure 3.9)

Figure 3.8

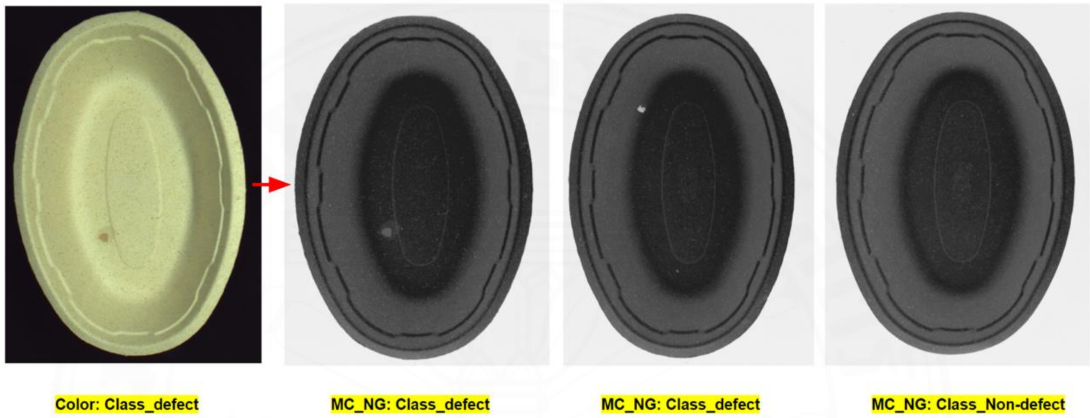
Color Histogram comparison between Negative Monochrome (NGMC) and RGB dataset



3.2.3.3 Storage saving, cut down image transfer time over the network, efficiently use computing resources and reduce model training time. Since at the same image resolution, the NGMC file is only 40% of RGB size.

Figure 3.9

Machine Vision configuration change between Color (RGB) and Negative Monochrome (NGMC)



3.3 Machine Learning Module

Our objective is to develop a conceptual framework for automated molded pulp defect detection system on the molded pulp packaging in the way that potentially serve as an industry-specific practical case study. The model must be able to classify whether the product images belong to a defect category or not. We adopt deep learning algorithm; Xception architecture with hyper-parameter tuning as our main goal. In comparison with supervised learning algorithms; SVM & Naive Bayes

3.3.1 Xception architecture

To choose the suitable value of hyper-parameters, it is important to build a robust deep learning model. For Xception model, the following technique is integrated to enhance performance of our classification models.

3.3.1.1 Image Resolution Tuning

To find optimal resolution that suited our model's application in order to reduce processing time and resource requirements, while maintain high training-testing accuracies. Our training resolution is varied in the range between 64×64 and 256×256 which is generally used to train CNNs as shown in Figure 3.10 & Figure 3.11.

Figure 3.10

Example of defect image on different resolution (a) 128 x 128 (b) 64 x 64

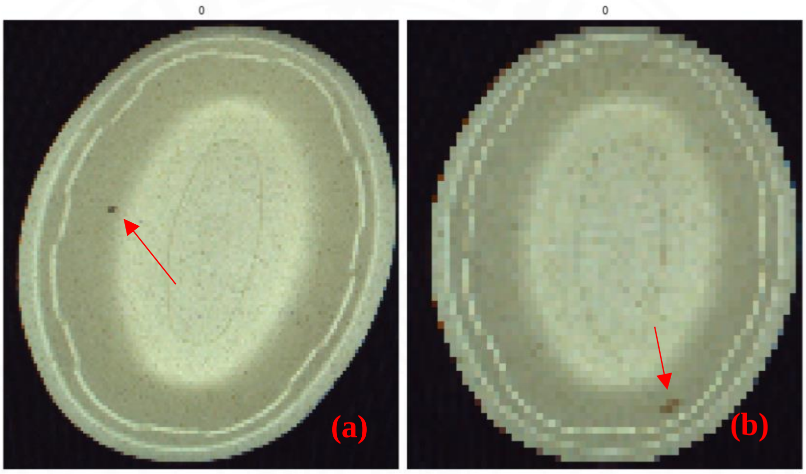
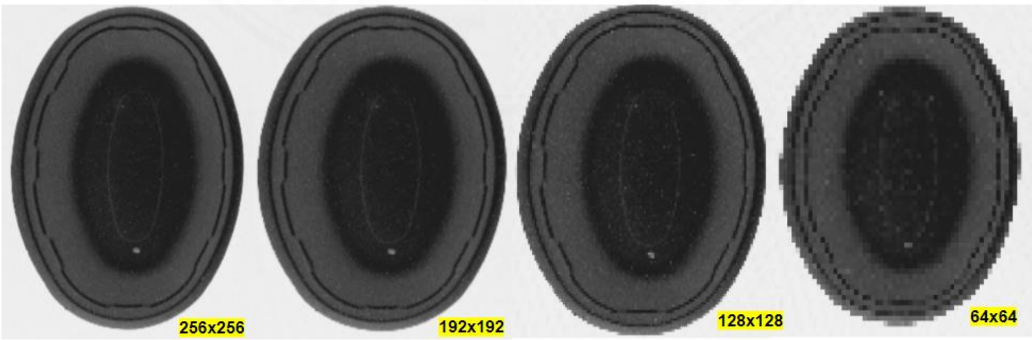


Figure 3.11

Example of image resolution varied in our research between 256x256 and 64x64



3.3.1.2 Learning Rate (LR) Tuning

Learning Rate (LR) Tuning: It is important to find the optimal learning rate for the model. In our research the Learning Rate was varied to include

0.004, 0.001(default), 0.0001 and 0.0004 to study how our defect detection model performance was affected by different Learning Rate. For Nadam optimizer used to train our Xception model, if we set learning rate too large, training process will very slowly as the model and making very tiny updates to the weights and lead to model not learning. However, once we set learning rate is too small, weight updated will be too large which led to undesirable divergent behavior in loss function and result in oscillate over training epochs. Therefore, we must not use too large or too small but find optimal learning rate.

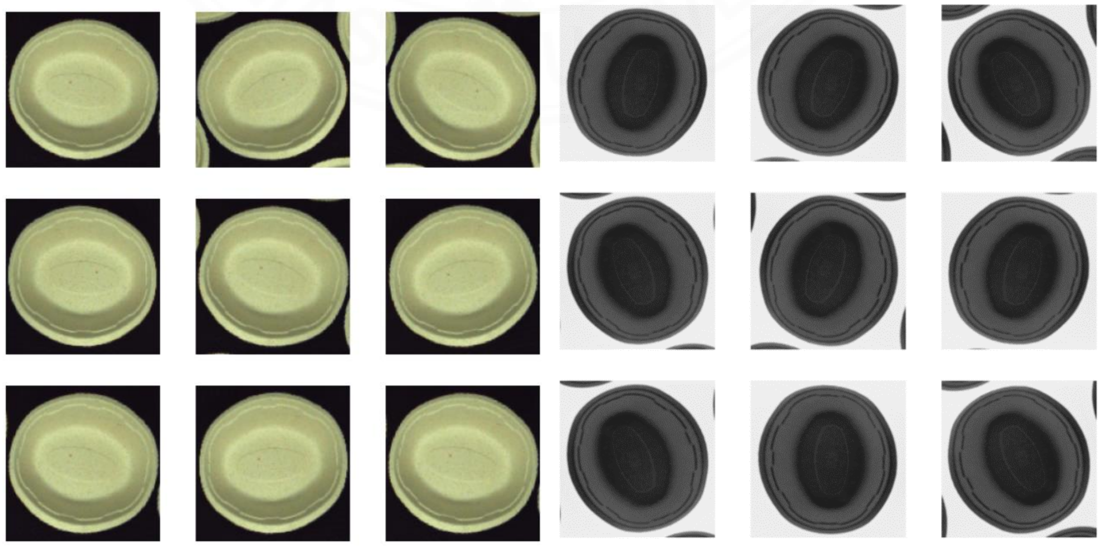
3.3.1.3 Batch Size (BS) Tuning

Batch size impacts the CNN training both in terms of the time to converge and the amount of overfitting. The purpose of our research is to find an appropriate range of batch size that yield best accuracy and generalization for our Xception model. The study is conducted by tuning the hyper-parameter batch size and observing its influences on training-testing accuracies. In our research the batch size was varied includes 1, 4, 8, 12, 16, 24, 32, 48 and 64.

In our research, we treat image augmentation and the optimizer algorithm as constant parameters. For augmentation of images, only flip and rotation techniques are implemented, as shown in Figure 3.12.

Figure 3.12

Implementation of Flipping and Rotation Augmentation on our research dataset



and only the Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer is implemented to acquire advantages from a fusion of the Adam optimizer and Nesterov Momentum algorithms, which can improve the speed of convergence and require the least exhaustive parameter tuning, which favors our focus on learning-rate (LR), image resolution, and batch size tuning.

Our Xception model uses the following settings: use the binary_crossentropy loss function, callbacks monitor max val_accuracy. To study the learning curve of the model across parameters tuning, we must set epochs at a number of hundred but not too high to prevent over-fitting, waste computing power and time. Hence, our models have decided to train for 350 epochs. After hyper-parameter tuning was completed, stratified 5-fold & 10-fold techniques were implemented in the final stage to train the best weight model. The performance of the model was measured for accuracy and precision. To avoid misclassifying a defective product as a non-defective one, we are focusing on the precision values of defective prediction. Our experiment results will be reported and discussed in chapters 4 and 5.

3.3.2 Supervised Learning

In comparison to the deep learning model - Xception architecture, we build supervised learning models - SVM and Naive Bayes. Our research implemented the Bag-of-Visual-Word (BoVW) approach to extract features from images for supervised learning, extract from original image pixel resolution of 1056 x 1520. Our training model setting was based on ORB_create() at default value and BoVW cluster size 500; the clustering size was decided to make visual words be able to represent all image patches but not too large to cause quantization artifacts or over-fitting.

All of the experiments are done on a standard Windows laptop computer using Google Colab Pro TPU - Google Compute Engine Python 3. Our experimental processes are shown in Chapter 4.

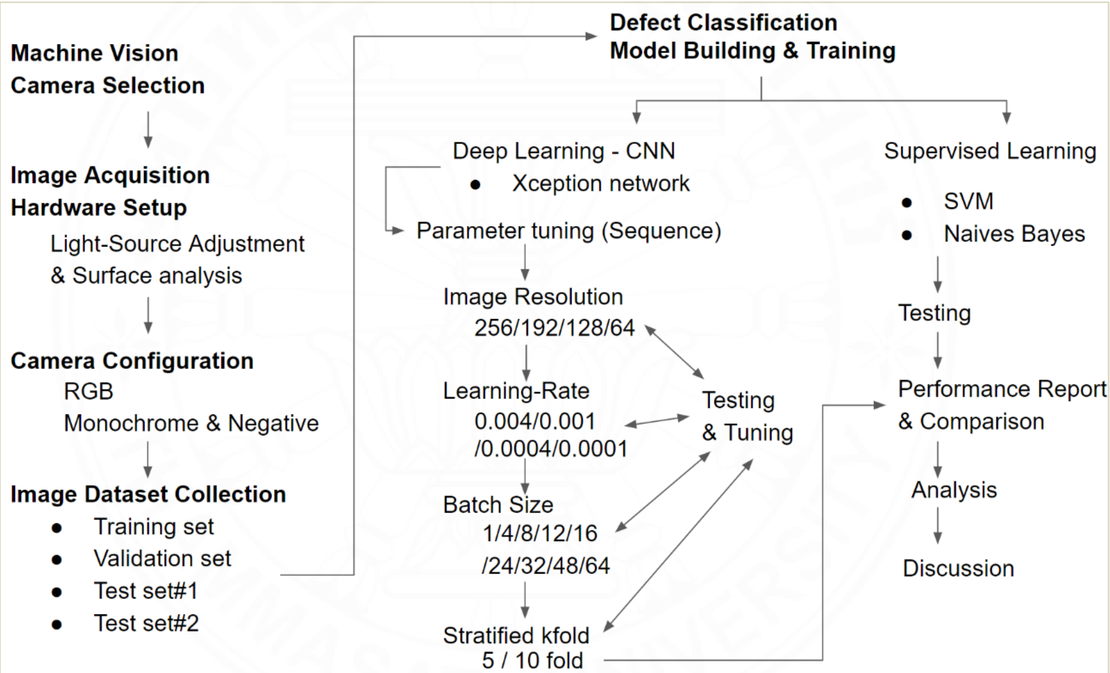
CHAPTER 4
RESULTS AND DISCUSSION

4.1 Design and Architecture of Research Experiment

Our experiment is described in detail as the flowchart in Figure 4.1.

Figure 4.1

The diagram presents the pipe-line of research experiment



Our experimental process consists of the following steps: The first step is to acquire training datasets via the image acquisition module using RGB and NGMC camera configurations. The next step is to explore suitable camera configurations by training the Xception model on RGB and NGMC datasets and comparing the model's performance, including accuracy, precision, and generalizability. Then we build the Xception model with hyper-parameter tuning, image resolution, learning-rate and batch size, respectively. Once the best model's hyper-parameter is found, the final model's performance is reported using Stratified k-fold cross-validation. Then, we build supervised learning models - SVM and Naive Bayes using ORB and BoVW feature extraction. Finally, all models' performances are reported and discussed.

4.2 Datasets Acquired by Image Acquisition Module

Samples of a molded pulp packaging product named "32oz burrito bowl" were imaged inside a light-box platform with circle-center LEDs to ensure uniform diffuse lighting. In comparison with RGB, the advantages of Negative Monochrome (NGMC) camera configurations are implemented to preprocess image datasets. The samples were collected from 4 different production lots using NGMC configuration, as listed in Table 4.1, from the molded pulp factory in Thailand, with an original image pixel resolution of 1056 x 1520. Since it is normal for product color to shift across the lots due to raw material color variation, as shown in Figure 4.2. Our classification models are generated using training & validation sets and then tested with test sets no.1 & no.2 to measure the model's generalizability across different lots.

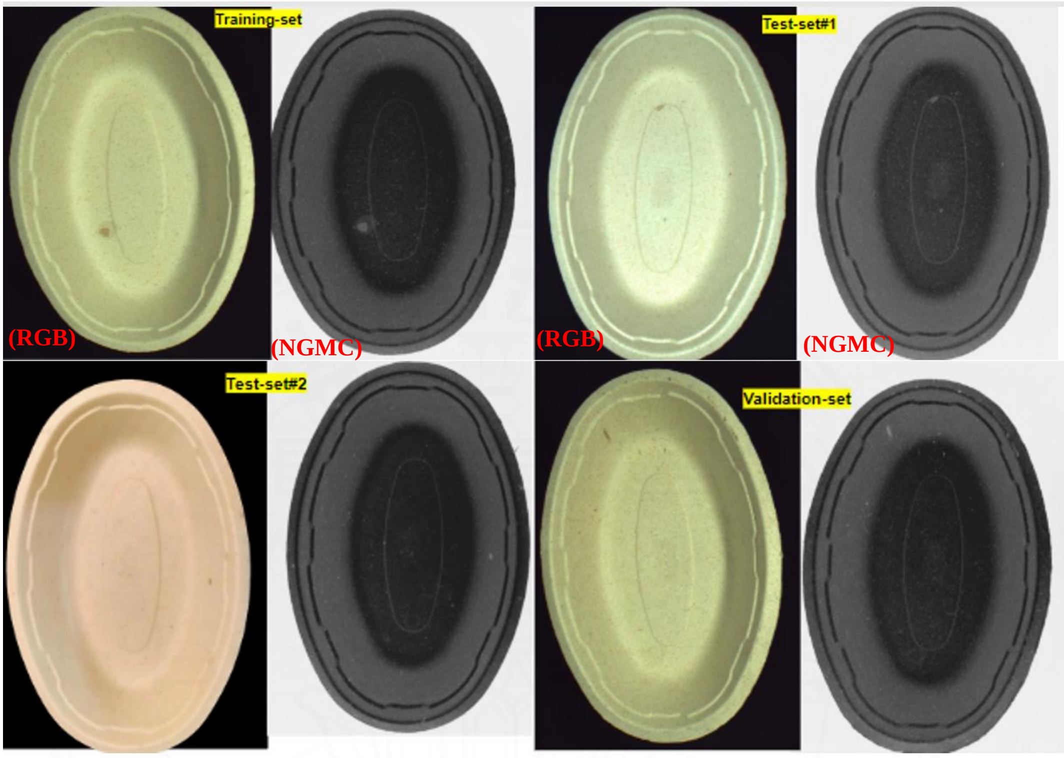
Table 4.1

Acquired Datasets for Model Building

Datasets Name	Sample
Training set	100 images with 50 defective and 50 non-defective
Validation set	40 images with 20 defective and 20 non-defective
Test set No. 1	57 images with 28 defective and 29 non-defective
Test set No. 2	60 images with 27 defective and 33 non-defective

Figure 4.2

Color comparison of different raw material lots between Color (RGB) and Negative Monochrome (NGMC)



4.3 Classifier Performance

Our research aims to classify whether the product images belong to a defect category or not. We adopt the deep learning algorithm; Xception as the base model. In comparison with supervised learning algorithms; SVM & Naive Bayes.

4.3.1 Deep learning - Xception architecture

4.3.1.1 Exploration on RGB Datasets

During the exploration stage of the Xception model, training, validation, and test set no.1 were implemented in the building and testing processes. Our initial model building is based on batch size 32, RGB image resolutions of 128x128 and 64x64 to minimize the computation resource. Overview of model training as shown in Figure 4.3.

Figure 4.3

Overview of Xception model training process (a) Import training-validation dataset and Batch size & Image resolution setting (b) Augmentation block (c) Xception model architecture building block (d) Model training block - setting and compile

The result of Xception model training and testing with RGB image resolution of 128x128 and 64x64 in comparison with NGMC are shown on Table 4.2

Table 4.2

Validation & testing results of Xception Model on RGB images resolution 128x128 and 64x64 in comparison with NGMC

Method				Dataset			
				Validation set		Test set No.1	
	Resolution	Image type	Learning-rate	Accuracy	Precision	Accuracy	Precision
Xception	128x128	RGB	0.001 (default)	88.00%	75.00%	52.00%	96.00%
	128x128	NGMC	0.001 (default)	95.00%	90.00%	84.00%	82.00%
	64x64	RGB	0.001 (default)	82.00%	75.00%	50.00%	100.00%

We have the following observations from the experimental results

1. Reducing image size could lower model fitting run-time.

2. When compare between RGB 128x128 and 64x64, the higher image resolution provider the better model accuracy.

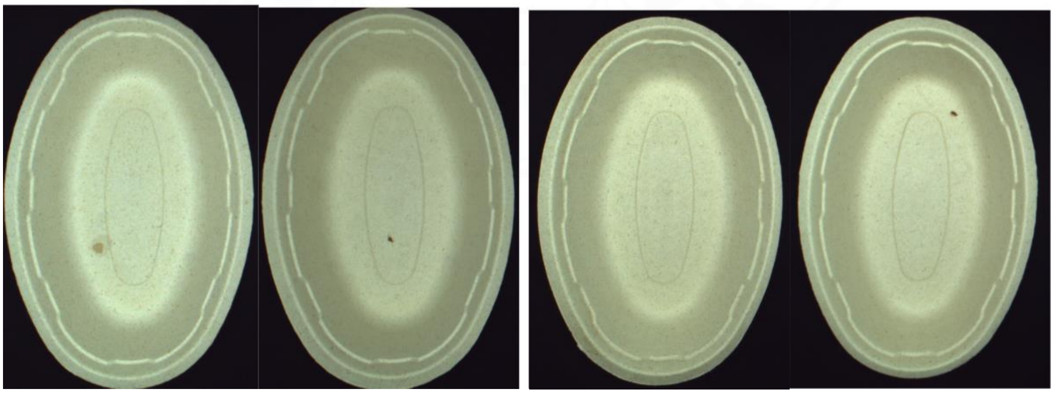
3. Since the model is weighted based on training and validation sets. When testing the RGB models on a new testing set no.1 from a different production lot, the model fails to predict the new dataset with only 50% accuracy. On the contrary, the NGMC model has higher generalizability, with 84% accuracy on test set no.1. The following hypotheses have been made:

3.1 The RGB model is over-fitted to the datasets or weighted against non-defect feature, hence low generalizability.

3.2 When comparing the RGB image - training and validation set with test set no.1, the products have surface fluctuations, especially by shade and color of raw material (Figure 4.4). The color of the raw material has an impact on the color of the product and defect. It contains excessive features and color channels that the model needs to learn to classify.

Figure 4.4

Example of images from training set, validation set and test set no.1



Training & Validation set **Test set no.1**

4.3.1.2 Model Performance on NGMC Datasets

Based on constraints discovered during the exploration stage of RGB datasets, the Negative Monochrome (NGMC) camera configurations will be applied in further stages to solve the color problem of products that change across raw material lots and to generalize the model learning pattern. The learning feature is reduced from 3 (RGB) to 1 color channel (NGMC), which decreases image file size and reduces computation resources. The following sub-sections describe the hyper-parameter tuning process used to build the Xception model.

(1) Image Resolution Tuning

To reduce processing time and resource requirements, the images are down-sampled to a fraction of the original resolution. However, extensive reduction of the image resolution eventually leads to the elimination and loss of important information in the image that is used for classification. Thus, first of all, we will find an optimal image with generalizability and classification performance.

Table 4.3

Validation & testing results of Xception Model on NGMC images with image resolution variation between 256x256 and 64x64

Method			Dataset					
	Resolution	Learning-rate	Validation set		Test set No.1		Test set No.2	
			Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Xception	256x256	0.001 (default)	97.50%	95.00%	96.49%	96.43%	88.33%	85.18%
	192x192	0.001 (default)	97.50%	100.00%	92.98%	96.43%	93.33%	92.59%
	128x128	0.001 (default)	95.00%	90.00%	84.21%	82.14%	78.33%	55.56%
	64x64	0.001 (default)	87.50%	85.00%	71.93%	60.71%	46.67%	7.41%

The results in Table 4.3 suggest that image quality affects the model's performance, where higher image resolutions result in higher accuracy and more generalization for deep learning models since the models take an image as input to learn visual information from the images. In contrast, the reduction of image resolution could eliminate the visual information (e.g., fine defects and contamination), which leads to the loss of important information in the image that is used for the classification and thus relatively poor performance at low resolutions. The 256x256 and 192x192 achieved highest accuracy when compare to others. However, when compared between 256x256 and 192x192 pixels, the lower input resolutions achieve better generalizability performance of the model across validation & test set. Based on theory (Thambawita, Strümke, Hicks, Halvorsen, Parasa, & Riegler, 2021), lower image resolution, which provides a smaller number of input variables or features, is often desirable in applications of deep architectures because there are fewer parameters that need to be learned and optimized, which could reduce the risk of model overfitting, in the case of 256x256 is more overfitted and least generalized compare to 192x192.

(2) Learning-Rate (LR) Tuning

After we achieve the optimal and best generalized model at an image resolution of 192 x 192, we will find the way to improve model performance via learning-rate tuning.

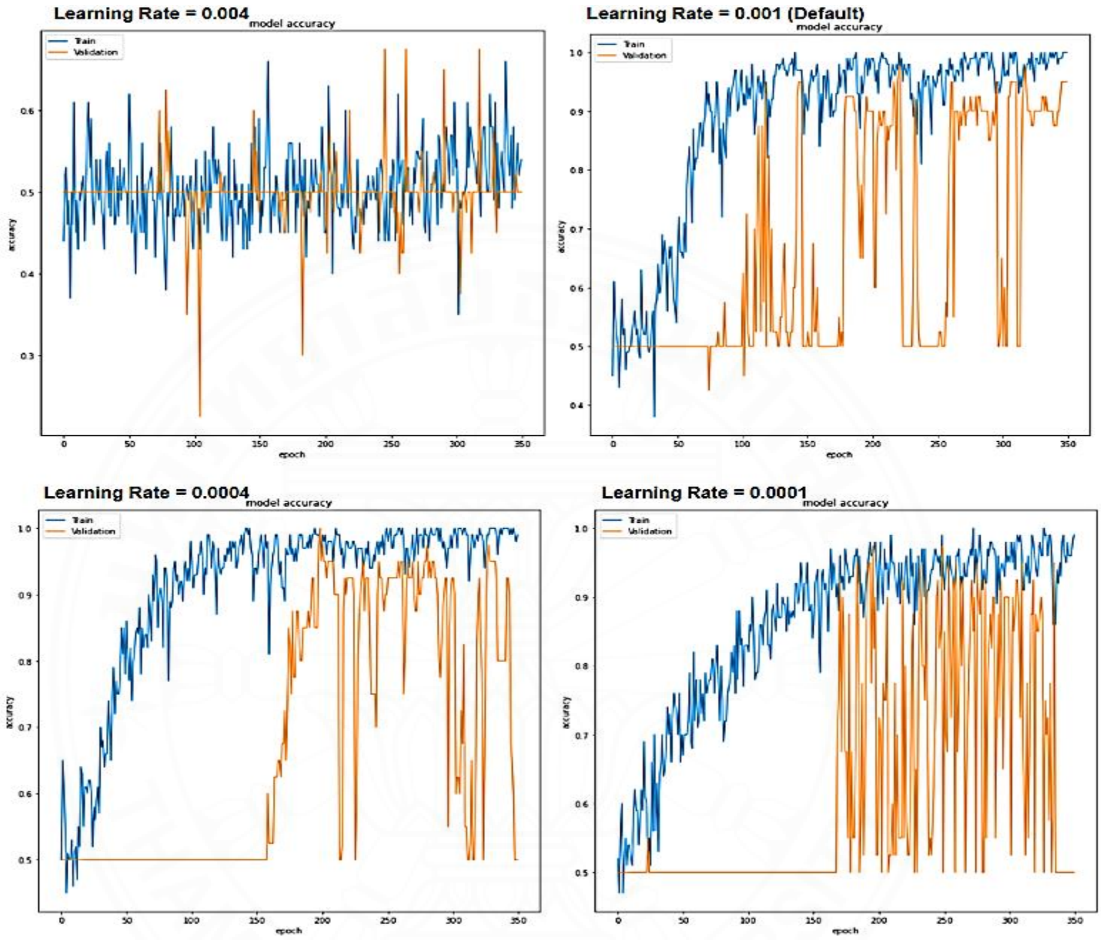
Table 4.4

Validation & testing results of Xception Model on NGMC images resolution 192x192 with learning-rate (LR) variation

Method			Dataset					
	Resolution	Learning-rate	Validation set		Test set No.1		Test set No.2	
			Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Xception	192x192	0.004	67.50%	45.00%	57.89%	25.00%	55.00%	11.11%
	192x192	0.001 (default)	97.50%	100.00%	92.98%	96.43%	93.33%	92.59%
	192x192	0.0004	100.00%	100.00%	87.72%	96.43%	96.67%	92.59%
	192x192	0.0001	97.50%	95.00%	80.70%	78.57%	63.33%	22.22%

Figure 4.5

Result comparison of Learning rate tuning for Xception model; Nadam optimizer on training and validation dataset NGMC image resolution 192x192



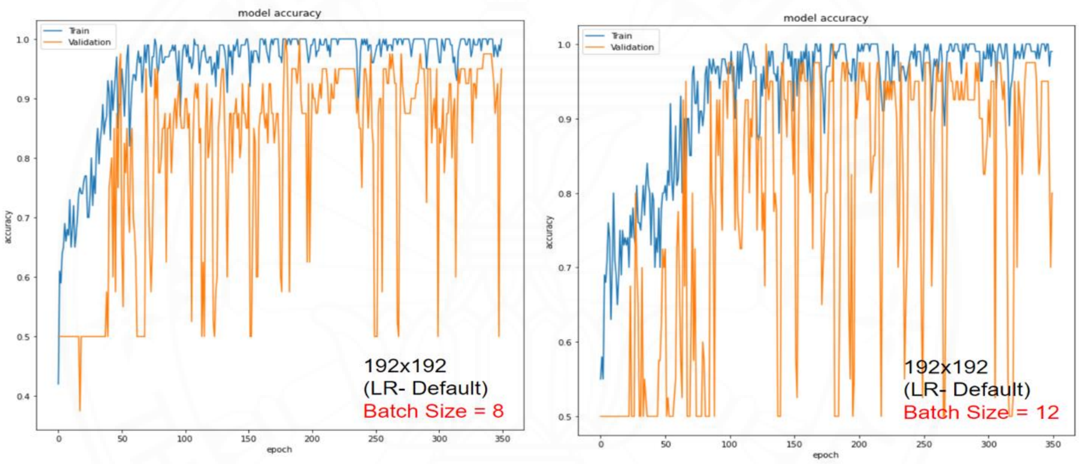
As can be seen from Figure 4.5 and Table 4.4, the generalization accuracy is quite poor for large learning rates LR-0004, and improves as the learning rate is reduced to a level of LR-0.001 to LR-0.0004. The learning rates from LR-0.0001 and smaller shows no improvement and have low generalization accuracy across test set no.1 & no.2. As observed from the result, if the learning rate is too large, accuracy will be poor due to Nadam optimizer making very tiny updates to the weights and lead to model not learning. In contrast when learning rate is too small, Nadam optimizer weight updated will be too large which could result in oscillate over training epochs. When compare between LR-0.001 and LR-0.0004, the LR-0.001 achieved better generalization accuracy with more than 92.98% across validation and set no.1 & no.2.

(3) Batch Size (BS) Tuning

Our batch size will be tuned based on the Xception model with optimal image resolution of 192x192 and best generalization accuracy of LR-0.001. To find an optimal batch size that yields reliable training results, the batch sizes are varied, as shown in Figure 4.6.

Figure 4.6

Training-validation performance curve comparison for batch size 8 and 12 of Xception model; Nadam optimizer LR-0.001 on training and validation dataset NGMC image resolution 192x192



Referring to the training-validation performance curve in Figure 4.7 and Figure 4.8, we observe that batch sizes between 8 and 48 yield high accuracies with a stable learning curve; when out of this range, the performance drops drastically. The batch sizes 8 and 12 achieved the highest accuracy, yielding the most stable and decent training curve.

Figure 4.7

Result comparison of batch size variation between 1 to 64 for Xception model; Nadam optimizer LR-0.001 on training and validation dataset NGMC image resolution 192x192

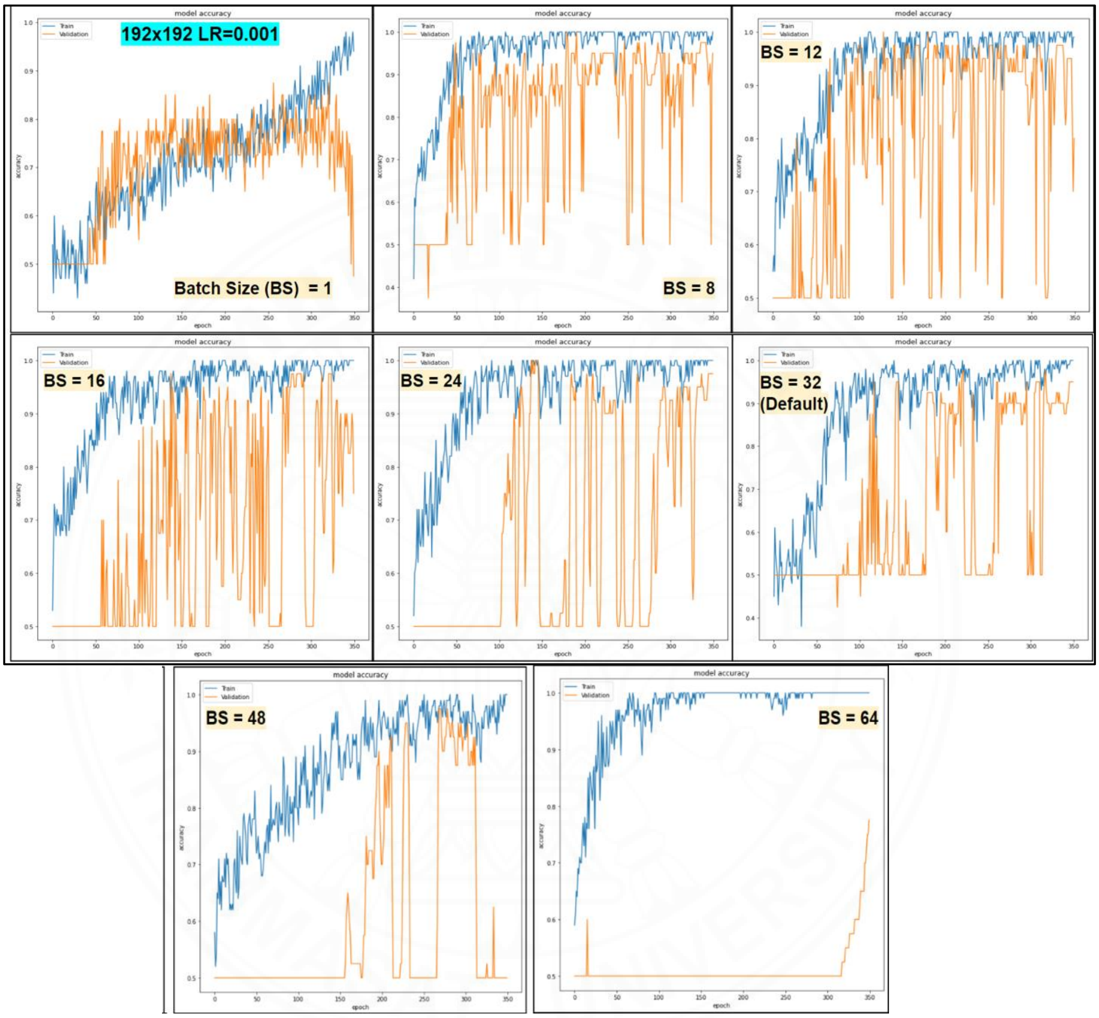


Table 4.5

Validation & testing results of Xception Model (LR-0.001) on NGMC images resolution 192x192 with batch size variation

Method				Dataset					
	Resolution	Learning-rate	Batch Size	Validation set		Test set No.1		Test set No.2	
				Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Xception	192x192	0.001 (default)	32	97.50%	100.00%	92.98%	96.43%	93.33%	92.59%
	192x192	0.001 (default)	12	100.00%	100.00%	96.61%	96.67%	86.66%	70.37%
	192x192	0.001 (default)	8	100.00%	100.00%	94.92%	100.00%	93.33%	88.89%

From Table 4.5, demonstrate that the batch size is vital to how the model learns the features and has an impact on the model's generalization performance on test sets no.1 & no.2. For high-complexity data of molded pulp packaging, the smaller batch size provides more gradient computation, yielding more stable and reliable training results. Thus, batch sizes of 8 provide high, stable accuracy across validation and test sets.

(4) Final Model Performance with Stratified k-fold

After the exploration and hyper-parameter tuning stages, the stratified k-fold (5-fold and 10-fold) will be applied to our most generalized model parameter, which is the Xception model; Nadam Optimizer (LR-0.001); NGMC image resolution 192x192 (batch size 8). As results are shown in Table 4.6, the Xception model trained with stratified k-fold has slightly better accuracy and generalizability when compared with normal training.

Table 4.6

Validation & Testing results comparison between Stratified k-fold and normal training on Xception Model (LR-0.001) NGMC images resolution 192x192 with batch size 8

Method				Dataset					
	Resolution	Learning-rate	Batch Size	Validation set		Test set No.1		Test set No.2	
				Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Xception	192x192	0.001 (default)	8 (Stratified 10-fold)	100.00%	100.00%	98.31%	96.67%	93.33%	85.19%
	192x192	0.001 (default)	8 (Stratified 5-fold)	97.50%	100.00%	93.22%	93.33%	95.00%	88.89%
	192x192	0.001 (default)	8	100.00%	100.00%	94.92%	100.00%	93.33%	88.89%
	192x192	0.001 (default)	12	100.00%	100.00%	96.61%	96.67%	86.66%	70.37%
	192x192	0.001 (default)	32	97.50%	100.00%	92.98%	96.43%	93.33%	92.59%

4.3.2 Supervised learning - SVM & Naive Bayes

In comparison with the Xception architecture, we build and train supervised learning SVM & Naive Bayes on NGMC image datasets of 1056 x 1520 pixel by implementing Bag-of-Visual-Word (BoVW) and ORB (Oriented Fast and Rotated Brief) for feature extraction. Overview of the supervised learning model building process as shown in Figure 4.8. The validation & testing results are shown in Table 5.7.

Figure 4.8

Overview of supervised learning - SVM & Naive Bayes training process

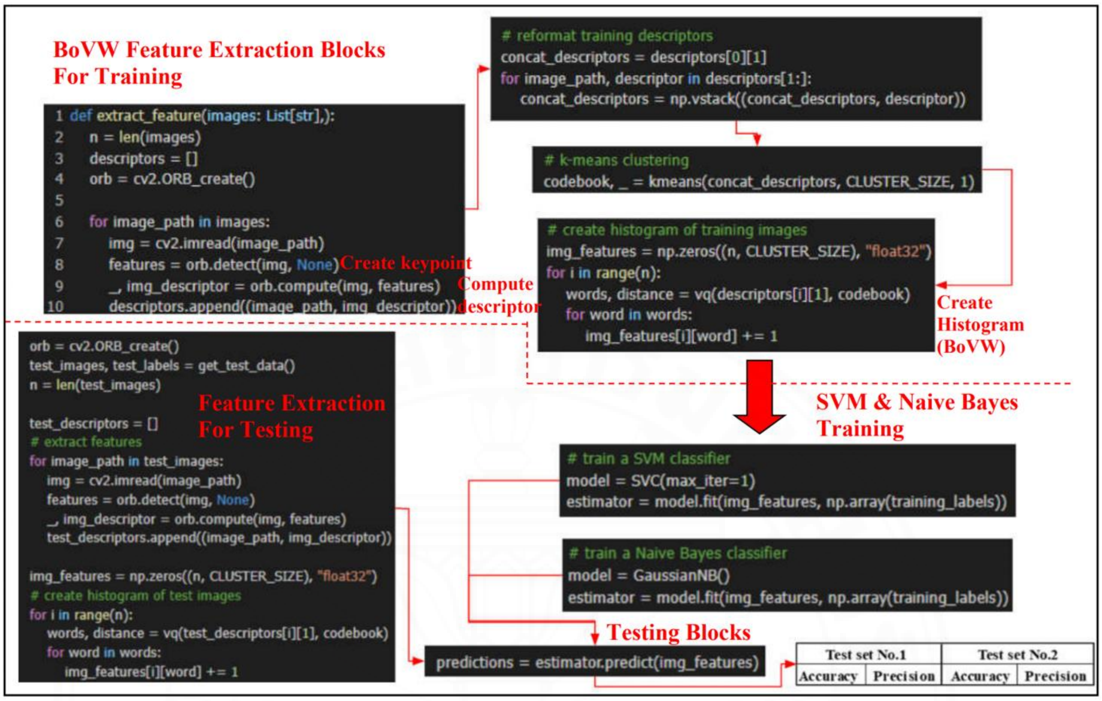


Table 4.7

Validation & testing results of Supervised Learning - SVM & Naive Bayes using BoVW and ORB for feature extraction on NGMC images dataset with Cluster size 500

Method	Dataset					
	Validation set		Test set No.1		Test set No.2	
	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
SVM	80.00%	83.33%	78.95%	75.00%	68.33%	61.11%
Naive Bayes	80.00%	75.00%	71.93%	66.67%	61.67%	54.35%

In our experiment, there are two main stages, as shown in Fig.46, the first is the training stage, and the second is the testing stage. In general, the first stage creates a BoVW visual vocabulary from training images. The information that was extracted in the first stage was used to train SVM and Naive Bayes supervised learning algorithm, which were then used to classify a new unlabeled image based on a bag of features created using the BoVW approach. Considering accuracy and precision between SVM and Naïve Bayes classifiers, the results obtained via SVM are the highest across datasets from different production lots. The model's performances are shown in Table 4.7. The following is our observation and hypothesis:

1. The keypoints and descriptors that the classification model learns during the training process can represent the defect features in validation and test set no.1, which makes the SVM predict with close to 80.00% accuracy in these two datasets. But when it comes to test set no.2, the accuracy and precision drop to 68.33% and 61.11%, respectively, which suggests the learned keypoints and descriptors are not generalized enough to classify defects in the new test set no.2. The potentials of the accuracy and precision drop are due to fluctuations in grain pulp fiber and non-repeating defect patterns, which were never learned by the classification model.

2. A supervised learning algorithm, including SVM and Naïve Bayes, requires pre-feature extraction from raw data via a human engineering algorithm. In our case, the ORB and BoVW techniques are used, which could be a drawback for supervised learning to learn the deep details of features. Unlike deep learning, the learning features are learned automatically through CNN from raw data, thus making it far superior to traditional machine learning in terms of feature learning and classification.

3. Comparing with deep learning, the supervised learning model has lower performance on both accuracy and precision and has the least generalizability across datasets from different production lots.

4.3.3 Model Performance Comparison

The models must be able to classify images accurately and must avoid misclassifying a defective product as a non-defective one; hence, we are focusing on the precision value of the defective category since the lower the precision value, the more overlooked cases cause more defective products to escape quality control to consumers, which could lead to consumer complaints and company losses. In Table 5.8 we report the results of all models based on NGMC image datasets.

Table 4.8

Result Comparison of Xception with hyper-parameter tuning and Supervised Learning - SVM & Naive Bayes on NGMC images Dataset

Method				NGMC Dataset					
	Resolution	Learning-rate	Batch Size	Validation set		Test set No.1		Test set No.2	
				Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Xception	256x256	0.001 (default)	32	97.50%	95.00%	96.49%	96.43%	88.33%	85.18%
	192x192	0.004	32	67.50%	45.00%	57.89%	25.00%	55.00%	11.11%
	192x192	0.001 (default)	32	97.50%	100.00%	92.98%	96.43%	93.33%	92.59%
	192x192	0.001 (default)	12	100.00%	100.00%	96.61%	96.67%	86.66%	70.37%
	192x192	0.001 (default)	8	100.00%	100.00%	94.92%	100.00%	93.33%	88.89%
	192x192	0.001 (default)	8 (Stratified 10-fold)	100.00%	100.00%	98.31%	96.67%	93.33%	85.19%
	192x192	0.001 (default)	8 (Stratified 5-fold)	97.50%	100.00%	93.22%	93.33%	95.00%	88.89%
	192x192	0.0004	32	100.00%	100.00%	87.72%	96.43%	96.67%	92.59%
	192x192	0.0001	32	97.50%	95.00%	80.70%	78.57%	63.33%	22.22%
	128x128	0.001 (default)	32	95.00%	90.00%	84.21%	82.14%	78.33%	55.56%
	64x64	0.001 (default)	32	87.50%	85.00%	71.93%	60.71%	46.67%	7.41%
SVM				80.00%	83.33%	78.95%	75.00%	68.33%	61.11%
Naive Bayes				80.00%	75.00%	71.93%	66.67%	61.67%	54.35%

For industrial applications, the deep learning algorithm - Xception using the Nadam optimizer (learning-rate 0.001) trained with Negative Monochrome (NGMC) preprocessed images at resolution 192x192 and batch size 8 achieved the most generalized results across validation and testing datasets, which suggest the model's capability to apply on products from different lots without problems caused by surface color fluctuation due to uncertain raw material shade and hue. Based on the processing time, the model built on an image resolution of 192 x 192 can make 60 predictions within 1s 900 ms, which has the potential to be utilized in an industrial application.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

Product quality control is an important factor in the long-term business growth of companies. Defect detection is also one of the most challenging parts of the manufacturing process, especially for molded pulp packaging, where the challenge is not only fluctuating grain surfaces and unstable product colors due to variation in raw material lots but also non-repeated defect sizes and patterns. While typical defect detection technologies, including X-ray detection, ultrasonic detection, and magnetic particle detection, are not suitable for this industry, the traditional molded pulp packaging manufacturing industry usually relies on the experience of quality inspectors to ensure the quality of products. Due to the uneven experience of inspectors and the limitations of human capabilities, this method has low efficiency, low accuracy, and poor real-time performance. With the further upgrading of the manufacturing industry in recent years, this method can no longer meet the requirements of high speed and high accuracy in modern industry.

This paper presents the end-to-end conceptual framework development for an automated molded pulp packaging defect detection using the conjunction of machine vision hardware and machine learning. The process includes machine vision camera selection, image acquisition platform setup in consideration of camera and lighting configuration, and machine learning model building based on the deep learning - Xception algorithm with hyper-parameter tuning in comparison with supervised learning - SVM & Naive Bayes implementing BoVW and ORB for feature extraction. We introduce the advantage of Negative Monochrome (NGMC) camera configurations to prevent color problems in products that change across raw material lots in order to generalize the model classification performance, which is limited when training the model on RGB datasets. This research shows the importance of using appropriate hyper-parameter optimization strategies in the deep learning model based Xception algorithm to obtain the best possible result, including image resolution tuning, learning rate tuning, and batch size tuning, respectively. The stratified k-fold was then implemented to train the model with optimized parameters in order to achieve the best final model performance. The experimental results show that the deep learning algorithm - Xception using the Nadam optimizer (LR-0.001) with Negative Monochrome (NGMC) resolution of 192x192 and batch size 8 provided optimized and best generalized performance across datasets from different production lots and have higher accuracy when compare to traditional supervised learning, which suggests that the robustness of our Xception model conceptual framework can make more than 60 predictions within 1s 900 ms and has the potential to be utilized in industrial.

5.2 Recommendation

There were still a limited number of features available to learn, and the dataset is still generally small, so the enhancement and expansion of the data set in the future will be particularly important to make the model improvements on a larger, structured dataset. At present, the application of machine learning in molded pulp packaging defect detection is still in the laboratory exploration stage, which requires more practical application in the actual detection environment of real-world manufacturing. Another limitation of our present work was that, owing to graphics processing unit memory constraints, as hardware advances make graphics processing units with larger amounts of random-access memory increasingly available, there is an opportunity for obtaining better performance from high image resolution models with larger batch sizes. For the introduction of Negative Monochrome (NGMC) configuration to improve the model generalizability is still in the initial stage, so there is still space for improvement and implementation in other applications. Since our conceptual framework is limited to an experimental hardware setup, there is still opportunity to integrate our defect detection conceptual framework with an on-field molded pulp packaging production. Our recommendations and challenges are as follows:

5.2.1 Real-time processing speed requirements

To build a model that is practical and real-time, we must aim for both classification accuracy and the cost of inference time. The main goal of on-field framework is to capture defect information globally with the highest accuracy and as fast as possible to satisfy real-time requirements of the production systems. As a result of our research's conceptual framework, the Xception model with suitable hyper-parameter configuration can achieve 1 s 900 ms inference time per 60 images in binary classification with more than 93.33% accuracy. When compared to current human manual inspection by well-trained operators who inspect products piece by piece, which requires 1 s per piece, the model outperforms humans by 97% in terms of speed. The model's performance shows the potential for replacing visual inspection jobs that were done by human workers with machine vision and machine learning in the near future.

5.2.2 The integration of machine vision and machine learning with molded pulp packaging production machine

First, to capture an image during production, the products come out of the machine in a cross-section area of 40 pieces per cycle, as shown in Fig. 49; hence, the machine vision installation location must be able to capture all over this area and transfer images to processing computer. As our research proposed on the machine vision hardware selection process, choosing the camera option and configuration that suit working applications and environments must be considered. The second is how the existing production machine control system responds to the classification result. Since the production machine is working on electric control, including a servo motor, solenoid valve and PLC, which is used to control pneumatic system, including a vacuum suction and air blow. The future work must perform on Python-based real-time control system to control the servo motors and pneumatic system of the production machine via read and write PLC process data using a python script in order classify product as in Fig. 50.

Figure 5.1

Machine's mould of "32 oz burrito bowl" cross-section area of 40 cavities

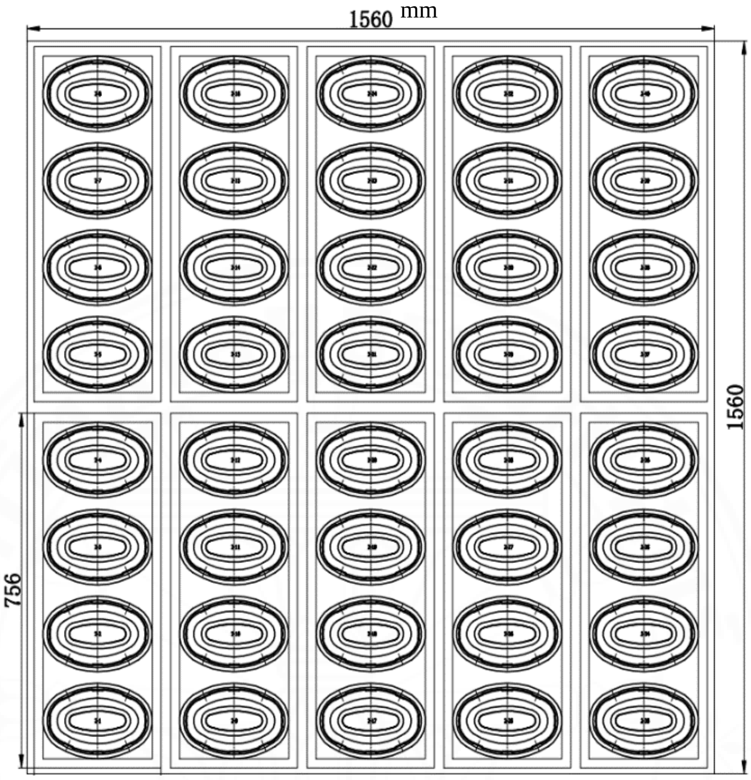
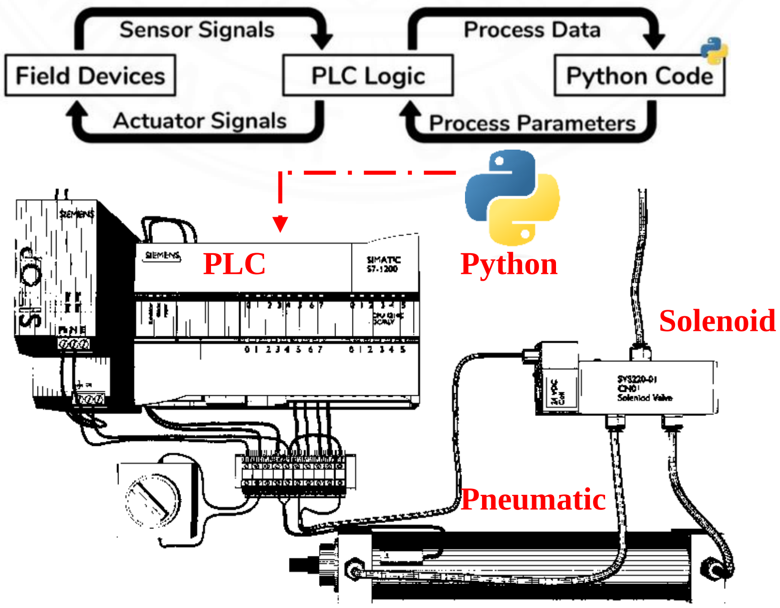


Figure 5.2

Two control loop of industrial Iot between Python-PLC and PLC-field device



Source: Neubert, J. (2023) and PLC Programming Tutorials Tips and Tricks. (2021)

REFERENCES

Articles

Albelwi, S., & Mahmood, A. (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. **Entropy, 19**(6), 242. https://doi.org/10.3390/e19060242

Boikov, A. V., Payor, V. A., Savelev, R. S., & Kolesnikov, A. (2021). Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. **Symmetry, 13**(7), 1176. https://doi.org/10.3390/sym13071176

Cadavid, J., Lamouri, S., Grabot, B., Pellerin, R., & Fortin, A. (2020). Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. **Journal of Intelligent Manufacturing, 31**(6), 1531–1558. https://doi.org/10.1007/s10845-019-01531-7

Gong, X., Bai, Y., Liu, Y., & Mu, H. (2020). Application of deep learning in defect Detection. **Journal of Physics: Conference Series, 1684**(1), 012030. https://doi.org/10.1088/1742-6596/1684/1/012030

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. **In International Conference on Machine Learning, 1**, 448–456.

Kabbai, L., Abdellaoui, M., & Douik, A. (2018). Image classification by combining local and global features. **The Visual Computer, 35**(5), 679–693. https://doi.org/10.1007/s00371-018-1503-0

Liu, Q., Wang, C., Li, Y., Gao, M., & Li, J. (2022). A Fabric Defect Detection Method Based on Deep Learning. **IEEE Access, 10**, 4284–4296. https://doi.org/10.1109/access.2021.3140118

Park, J., Kwon, B., Park, J. Y., & Kang, D. (2016). Machine learning-based imaging system for surface defect inspection. **International Journal of Precision Engineering and Manufacturing-Green Technology, 3**(3), 303–310. https://doi.org/10.1007/s40684-016-0039-x

Ren, Z., Fang, F., Yan, N., & Wu, Y. (2021). State of the Art in Defect Detection Based on Machine Vision. **International Journal of Precision Engineering and Manufacturing-Green Technology, 9**(2), 661–691. https://doi.org/10.1007/s40684-021-00343-6

Sabottke, C. F., & Spieler, B. (2020). The Effect of Image Resolution on Deep Learning in Radiography. **Radiology, 2**(1), e190015. https://doi.org/10.1148/ryai.2019190015

Sanghadiya, F., & Mistry, D. (2015). Surface Defect Detection in a Tile using Digital Image Processing: Analysis and Evaluation. **International Journal of Computer Applications, 116**(10), 33–35. https://doi.org/10.5120/20375-2592

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. **Journal of Big Data, 6**(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Sultani, Z. N., & Dhannoon, B. N. (2021). Modified Bag of Visual Words Model for Image Classification. **Al-Nahrain Journal of Science, 24**(2), 78–86. https://doi.org/10.22401/anjs.24.2.11

Thambawita, V., Strümke, I., Hicks, S. A., Halvorsen, P., Parasa, S., & Riegler, M. (2021). Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images. **Diagnostics, 11**(12), 2183. https://doi.org/10.3390/diagnostics11122183

Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., & Tang, S. (2020). Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges. **Materials, 13**(24), 5755. https://doi.org/10.3390/ma13245755

Zhao, W., Chen, F., Huang, H., Li, D., & Cheng, W. (2021). A New Steel Defect Detection Algorithm Based on Deep Learning. **Computational Intelligence and Neuroscience, 2021**, 1–13. https://doi.org/10.1155/2021/5592878

Electronic Media

Advanced Illumination. (2023). **A Practical Guide to Machine Vision Lighting - Advanced Illumination.** https://www.advancedillumination.com/a-practical-guide-to-machine-vision-lighting/

Arshad, S. N., Obaid, I., Gull, R., & Shahzad, M. (2022). **Steel Defect Classification Using Machine Learning.** https://ieeexplore.ieee.org/document/9721728

Bayes Theorem - Formula, Statement, Proof | Bayes Rule. (n.d.). **Cuemath.** https://www.cuemath.com/data/bayes-theorem/

Brownlee, J. (2021). **Gradient Descent Optimization With Nadam From Scratch.** https://machinelearningmastery.com/gradient -descent-optimization-with-nadam-from-scratch

Burresi, G., Lorusso, M., Graziani, L., Comacchio, A., Trotta, F., & Rizzo, A. (2021). **Image-Based Defect Detection in Assembly Line with Machine Learning.** https://ieeexplore.ieee.org/document/9460291

Chernytska, O. (2023). **Complete Guide to Data Augmentation for Computer Vision. Medium.** https://towardsdatascience.com /complete-guide-to-data-augmentation-for-computer-vision-1abe4063ad07

Chollet, F. (2017). **Xception: Deep Learning with Depthwise Separable Convolutions.** https://ieeexplore.ieee.org/document/8099678

Chouinard, J. (2023). **What is a lens optical format? Can I use any machine vision camera with any format? NOT! 1stVision Inc. - Machine Vision Articles.** https://www.1stvision.com/machine-vision-solutions/2017/08/how-does-a-lens-optical-format-relate-to-machine-vision-cameras.html

CIS520 Machine Learning | Lectures / Precision Recall. (n.d.). **Precision Recall.** https://alliance.seas.upenn.edu/~cis520/dynamic/2017/wiki/index.php?n=Lectures.PrecisionRecall

Concept of Minimum Detectable Object Size | Machine Vision Basics | KEYENCE America. (n.d.-b). https://www.keyence.com/ss/products /vision/visionbasics/basic/practice02/

Cross-validation: evaluating estimator performance. (n.d.). **Scikit-learn.** https://scikit-learn.org/stable/modules/cross_validation.html

Dang, D., Wang, J., Lee, J., & Wang, C. (2021). **Defect Classification System for Ski Goggle Lens.** https://ieeexplore.ieee.org/document/9651116

Lin, R. (2022). **Analysis on the Selection of the Appropriate Batch Size in CNN Neural Network.** https://doi.org/10.1109/mlke55170.2022.00026

Lo, W. H., Yang, X., & Wang, Y. (2019). **An Xception Convolutional Neural Network for Malware Classification with Transfer Learning.** https://doi.org/10.1109/ntms.2019.8763852

Luckert, M. (2016). **Using Machine Learning Methods for Evaluating the Quality of Technical Documents.** https://www.semanticscholar.org/paper/Using-Machine-Learning-Methods-for-Evaluating-the-Luckert-SchaeferKehnert/87acf33976 71166fe99ffdc29acd977132749662

Masters, D., & Luschi, C. (2018). **Revisiting Small Batch Training for Deep Neural Networks. ResearchGate.** https://www.researchgate.net/publication /324690431_Revisiting_Small_Batch_Training_for_Deep_Neural_Networks

MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. (n.d.). http://yann.lecun.com/exdb/mnist/

Molded Pulp Packaging Market Size, Share & Trends Analysis Report By Source (Wood Pulp, Non-Wood Pulp), By Molded Type (Thick Wall, Thermoformed, Transfer, Processed), By Product, By Application, By Region, And Segment Forecasts, 2023 - 2030. (n.d.). https://www.grandviewresearch.com/industry-analysis/molded-pulp-packaging-market

Neubert, J. (n.d.). **MQTT: A Pythonic introduction to the protocol for connected devices - Jonas Neubert.** https://jonasneubert.com/talks/pycon2023.html

Ngo, T.D., Abdukhakimov, A., & Kim, D. (2019). **Long-Range Wireless Tethering Selfie Camera System Using Wireless Sensor Networks.** https://ieeexplore.ieee.org/document/8789410

Optimization Algorithms - A Brief Overview:- · GSoC'18 @ CERN. (2018, May 16). https://www.sravikiran.com/GSOC18//2018/05/16/optimizers/

Pandian, S. (2022). **K-Fold Cross Validation Technique and its Essentials.** https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross- validation-technique-and-its-essentials

PLC Programming Tutorials Tips and Tricks. (2021). **How to operate a Pneumatic Cylinder with PLC ? PLC programming tutorial for beginners** [Video]. YouTube. https://www.youtube.com/watch?v=DN9cnTTqXPA

Prasitmeeboon, P., & Yau, H. (2019). **Defect Detection of Particleboards by Visual Analysis and Machine Learning.** https://ieeexplore.ieee.org/document/8802526

Pulp. (2018). **How to make pulp with bagasse?.** http://www.paper pulping.com /news/How-to-make-pulp-with-bagasse.html

Selecting the Correct Camera | Machine Vision Basics | KEYENCE America. (n.d.). https://www.keyence.com/ss/products/vision/visionbasics/tips/primer1/

Shakeri, M., Ariannejad, M., Sedaghati, N., & Amin, M. S. (2012). **Advanced cmos based image sensors. ResearchGate.** https://www.researchgate.net /publication/ 288537030_Advanced_cmos_based_image_sensors

Shenzhen Mind vision technology (n.d.). https://www.mindvision.com.cn/jtxx/list_108.aspx?lcid=18&lcids=2414]

Team, K. (n.d.). **Keras documentation: Keras Applications.** https://keras.io/applications/#xception

Vandendorpe, A. (2022). **How to Choose a Camera for Machine Learning - ML6team. Medium.** https://blog.ml6.eu/how-to-choose-a-camera-for-ml-e2a1819f37e0

Wikiwand - Support Vector Machine. (n.d.). **Wikiwand.** https://www.wikiwand.com/de/Support_Vector_Machine

Zaborowska, Ł. (2023). **Focal Length Calculator. omnicalculator.** https://www.omnicalculator.com/other/focal-length

APPENDICES

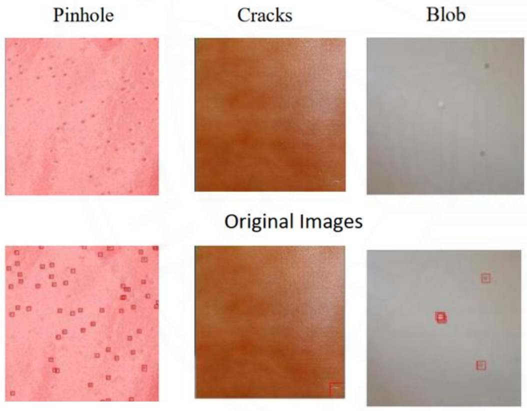
APPENDIX A

Defect detection using image processing technique

To find defects, various image processing techniques can be applied on an image of target object such as tile, steel slab and fabric. While inspecting an object, we may see certain defects in the surface of an object. In case of tile, we could find blob, pinhole, variation in color of a tile, crack, and chip offs in a tile, pattern mismatch, scratches etc. as shown in Fig.51., which can apply some image processing steps to find these defects but each defect, different technique is applicable to gain efficient result (Sanghadiya, & Mistry, 2015).

Figure A.1

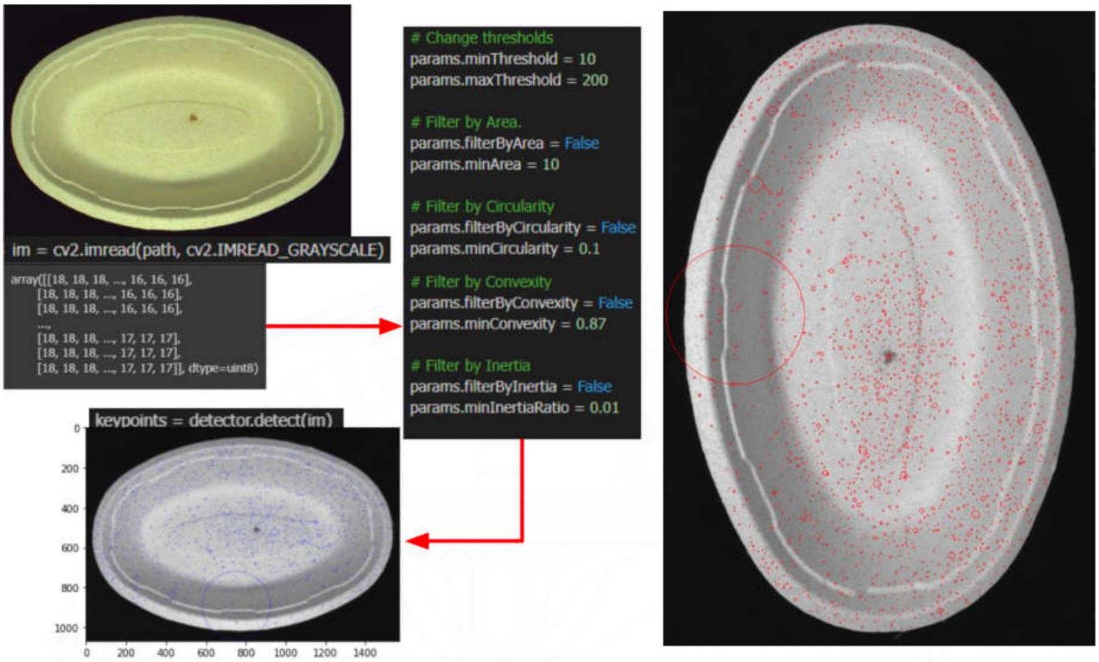
Experimental results of detection of various types of defects using different type of image processing and its drawback



Defect	Technique	Drawback
Blob	BlobDetector and Segmentation using particular threshold value	Difficult to find particular threshold value.
Pinhole	Thresholding	Difficulty in finding threshold value.
Cracks	Local maxima & Wigner distribution	More computation time

Figure A.2

Implementation of OpenCV BlobDetector on molded pulp packaging



For our molded pulp packaging, we implemented OpenCV BlobDetector to find its potential in defect detection tasks on our target object. The experimental and result are shown in Fig.52. The process includes (1) converting the source image to binary images (2) extract connected components. (3) group centers by their coordinates; close centers form one group that corresponds to one blob. (4) estimate the final centers of blobs and their radiuses, then return the locations and sizes of keypoints. We following constraints of image process for defect detection task are observed:

1. Since molded pulp packaging has obstacles such as surface fluctuation by shade, grain of pulp fiber and non-repeating defect patterns, it is difficult to find a particular parameter value on BlobDetector so that the algorithm can detect the geometric pattern of defects.

2. Some image processing methods for detection of object give better results but consume more time which is not acceptable in real-time environment.

3. No such algorithm can find all the defects at a time.

According to the mentioned drawbacks of image processing for defect detection, it suggests the inapplicability of building an automated molded pulp packaging defect detection conceptual framework based on image processing techniques.

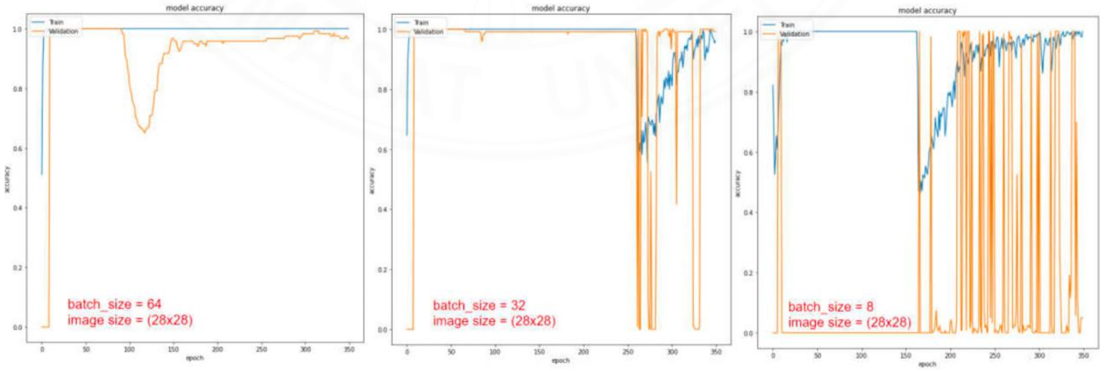
APPENDIX B

Xception model performance on the MNIST dataset

In comparison to the high complexity of the molded pulp packaging dataset, we tested our Xception model with batch size variation on the MNIST dataset. The MNIST database (Modified National Institute of Standards and Technology database (LeCun, Cortes and Burges, n.d.).) is a large database of handwritten digits that is widely used for training and testing in the field of machine learning. Due to its low complexity, the MNIST dataset is easier to fit one batch worth of training data in memory when using a GPU or TPU and requires low computation power when compared to the defect classification model of molded pulp packaging. In this research, our MNIST dataset contains 400 files, which belong to two classes: 280 files for training and 120 files for validation. Our classification model is based on deep learning using the Xception architecture with a learning rate of 0.001 (default), the Ndam optimizer, and batch sizes of 64, 32, and 8. The model's learning curve performance is shown in Fig 53.

Figure B.1

Training-validation performance curve comparison for batch sizes of 64, 32, and 8 of Xception model; Nadam optimizer LR-0.001 on MNIST dataset



The batch size represents the number of training samples that will be used during the training in order to make one update to the network parameters. The performance curves show that it is feasible and effective to use a large batch size for complexity and a small sample size. The large batch size of 64 has lower fluctuations and is more generalized when compared to batch sizes 32 and 8. Unlike the performance curves of molded pulp packaging from our research (Fig. 47), the lower batch sizes of 8 and 12 are preferred for larger and more complex datasets. The molded pulp packaging dataset is an extensive database; feeding all the data into the network at once could lead to an explosion of memory, so finding an optimal batch size is important for the deep learning model training process.

BIOGRAPHY

Name	Rattana Ceesay
Educational Attainment	Academic Year 2017: Bachelor of Engineering Chemical Engineering, Faculty of Engineering, Kasetsart University Thailand

Publications

Rattana Ceesay, Thapana Boonchoo and Prapaporn Rattanatamrong. “Machine Learning Approaches for Quality Control in Pulp Packaging Manufacturers” paper presented at the 20th International Joint Conference on Computer Science and Software Engineering (JCSSE2023), Phitsanulok, Thailand, 385-390, 2023