Faculty of Electrical Technology and Engineering

DEVELOPMENT OF THE VISUAL MONITORING SYSTEM BASED ON AN INDUSTRIAL AUTOMATION SYSTEM (GLASS INDUSTRY) BY USING VB.NET

PAVITHRAN A/L GUNASEELAN

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotic) with Honours

2023

DEVELOPMENT OF THE VISUAL MONITORING SYSTEM BASED ON AN INDUSTRIAL AUTOMATION SYSTEM (GLASS INDUSTRY) BY USING VB.NET

PAVITHRAN A/L GUNASEELAN

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology with Honours

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Development Of The Visual Monitoring System Based On An Industrial Automation System (Glass Industry) By Using VB.NET

Sesi Pengajian : 2023/2024

Saya PAVITHRAN A/L GUNASEELAN mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
✓	**TIDAK TERHAD**	

Disahkan oleh:

SHAHRIDIN BIN ZAKARIA
Pensyarah
Fakulti Teknologi Kejuruteraan Elektrik

(TANDATANGAN PENULIS) (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap:

NO 6, Jalan dato kaya kecil 4b, Taman Aman, Kapar 42200, Selangor Darul Ehsan.

Tarikh: 4/1/2024 Tarikh: **15/2/2024**

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development Of The Visual Monitoring System Based On An Industrial Automation System (Glass Industry) By Using VB.NET" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : PAVITHRAN A/L GUNASEELAN

Date : 14/1/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology with Honours.

Signature : ....

Supervisor Name : TS . SHAHRUDIN BIN ZAKARIA

Date : **15/2/2024**

Signature : ....

Co-Supervisor Name (if any) : ....

Date : ....

DEDICATION

To my beloved father, GUNASEELAN A/L TETTIAN, and mother, TAMIL CHELVI A/P RAJOO, who have always been my pillars of strength and support throughout my life, your unwavering love and guidance have been instrumental in shaping me into the person I am today. I would also like to dedicate this project to my dearest friend GUGAN A/L MANIKAM who have been my constant companion and confidant. Your encouragement and support have been invaluable to me, and I am grateful for your unwavering belief in me.

ABSTRACT

The project " Development Of The Visual Monitoring System Based On An Industrial Automation System (Glass Industry) By Using VB.NET" aims to gather detailed information, minimize human intervention to enhance quality control, safety and operational efficiency in the glass manufacturing sector. The glass manufacturing industry requires constant vigilance for maintaining quality and operational efficiency. Manual monitoring becomes challenging and error-prone, particularly in high-speed production environments. This project aims to address these challenges by implementing automated defect monitoring on the glass bottle manufacturing floor. The core of the system integrates cutting-edge computer vision capabilities with real-time control mechanisms in a manufacturing environment. The proposed system features a user-friendly Human-Machine Interface (HMI) through Visual Basic (VB.NET) and employs seamless interaction with image recognition techniques utilizing Convolutional Neural Networks (CNNs) deep learning algorithm for glass bottle monitoring. The methodology includes the design and implementation of the visual monitoring and inspection system, with an emphasis on the system architecture, hardware components, and software implementation. The results showcase the systems commendable performance in distinguishing between proper and defective glass bottles with high accuracy on the conveyor belt. The pre-trained predictive model, powered by Convolutional Neural Networks (CNNs), proves its efficacy in efficiently categorizing bottles, contributing to a streamlined defect detection process. Moreover, the real-time colour recognition feature enhances analytical capabilities by extracting and processing colour details from frames. The VB.NET interface serves as a comprehensive tool for generating detailed reports, offering insights into the monitoring process with statistics on proper and defective bottles. This empowers users to make informed decisions and optimize processes. Overall, by leveraging a camera sensor and seamlessly integrating advanced CNN-based monitoring with a versatile VB.NET interface, this system transforms manufacturing quality control, ensuring real-time, automated monitoring, and efficient segregation of defective products. Ultimately, this enhances overall production efficiency and product quality in a streamlined process.

ABSTRAK

Projek yang bertajuk "Pembangunan Sistem Pemantauan Visual Berdasarkan Sistem Automasi Industri (Industri Kaca) Dengan Menggunakan VB.NET" bertujuan untuk mendapatkan maklumat terperinci, mengurangkan campur tangan manusia, dan meningkatkan kawalan kualiti, keselamatan, dan kecekapan operasi dalam sektor pembuatan kaca. Keperluan yang berterusan untuk berjaga-jaga dalam mengekalkan kualiti dan kecekapan operasi dalam industri pembuatan kaca, terutamanya dalam persekitaran pengeluaran berkelajuan tinggi, diatasi dengan pelaksanaan pemantauan kerosakan automatik di lantai pembuatan botol kaca. Sistem ini menggabungkan keupayaan penglihatan komputer terkini dengan mekanisme kawalan secara masa nyata, menampilkan Antara Muka Mesin-Manusia (HMI) yang mesra pengguna melalui VB.NET. Penggunaan algoritma pembelajaran mendalam Convolutional Neural Networks (CNNs) membolehkan pemantauan botol kaca yang cekap. Metodologi menekankan senibina sistem, komponen perkakasan, dan pelaksanaan perisian. Keputusan menunjukkan prestasi yang memuaskan dalam membezakan antara botol kaca yang betul dan yang rosak dengan tepat di atas penyampai berjalan. Model prediktif pra-latih, dikuasakan oleh CNNs, dengan cekap mengategorikan botol, menyempurnakan proses pengesanan kerosakan. Ciri pengenalan warna secara masa nyata meningkatkan keupayaan analisis dengan mengekstrak dan memproses butiran warna dari bingkai. Antara Muka VB.NET berfungsi sebagai alat komprehensif, menghasilkan laporan terperinci dengan statistik tentang botol yang betul dan yang rosak, memberi kuasa kepada pengguna untuk membuat keputusan yang berinformasi dan mengoptimumkan proses. Dengan memanfaatkan sensor kamera dan mengintegrasikan pemantauan canggih berasaskan CNN dengan Antara Muka VB.NET yang serba guna, sistem ini mengubah kawalan kualiti pembuatan, memastikan pemantauan automatik secara masa nyata, dan pemisahan yang cekap bagi produk yang rosak. Pada keseluruhannya, ini meningkatkan kecekapan pengeluaran dan kualiti produk secara menyeluruh dalam proses yang efisien.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, TS . SHAHRUDIN BIN ZAKARIA for his precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for the financial support through this semester which enables me to accomplish the project. Not forgetting my fellow classmates, MUHAMMAD HARITH BIN MOHAMMAD NIZAM and MUHAMMAD RIDZWAN HAKIM BIN REZALI for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents, and family members for their love and prayer during the period of my study. Finally, I would like to thank all the staffs at the staffs at Universiti Teknikal Malaysia Melaka (UTeM), fellow friends and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

TABLE **TITLE** **PAGE**

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

The reader will get an overview of the project in this chapter. It includes the study's history, project purpose, problem statement, and scope. It explains the inspiration for this study as well as the factors that led to it.

1.1 Background

The glass manufacturing industry plays a pivotal role in the global economy, contributing significantly. However, it grapples with inherent risks in critical processes like glass melting and shaping, impacting worker safety and environmental concerns. This project's primary objective is to address these challenges by advancing automated monitoring on the manufacturing floor. The aim is to acquire more detailed information while minimizing the need for extensive human interaction, thus enhancing overall safety and operational efficiency in the glass manufacturing sector. Glass manufacturing demands continuous attention to quality and operational efficiency. Manual monitoring is time-consuming and error-prone, especially in high-speed production. To tackle this, the project introduces a revolutionary image recognition and deep learning approach with Convolutional Neural Networks (CNNs) for automatic glass bottle defect detection. CNNs swiftly and accurately analyze bottle images, providing real-time anomaly diagnosis. This precision forms the system's cornerstone. Additionally, the system integrates a real-time colour recognition feature, enhancing its analytical capabilities. The colour recognition module extracts and identifies colours from captured frames, offering an added dimension to the defect categorization process. Complementing CNNs with a user-friendly Visual Basic (VB.NET) interface, vital for bidirectional communication between CNN models and control mechanisms. The interface provides real-time inspection findings and immediate control over conveyor operations. Beyond defect control, the system aims to redefine quality assurance in manufacturing, revolutionizing the interplay between powerful neural networks and an interactive control interface. This approach enhances overall efficiency and product quality in the glass manufacturing process for a safer and streamlined operational environment.

1.2 Problem Statement

The glass bottle manufacturing industry is a critical sector that produces various types of bottles for different purposes. The production process involves the use of high-temperature ovens and molds, which are potentially dangerous to human operators. The industry requires a automated visual monitoring system that can ensure the smooth and safe operation of the manufacturing process with minimal human intervention.

The glass bottle manufacturing industry's monitoring system is insufficient in detecting and addressing production deviations in real-time. The manual monitoring process is highly inefficient and can lead to costly production delays, quality issues, and safety hazards for workers. These critical processes expose human workers to various occupational hazards that could affect their health and safety.

Traditional solutions, involving sensors and alarms, have fallen short in minimizing human intervention and enhancing the production process's efficiency.line. However, these solutions were not effective in reducing the human intervention in the production process.

The envisioned solution involves creating an advanced visual monitoring system using VB.NET, strategically integrating cutting-edge image recognition techniques with

Convolutional Neural Networks (CNNs). By incorporating cameras, this system excels in real-time detection of glass bottle defects, providing immediate data visualization and alerts for operators. This innovative approach significantly reduces human intervention, fostering a safer operational environment, mitigating errors and accidents, and ultimately elevating production quality and efficiency in the glass bottle manufacturing process.

1.3 Societal and Global Implications of Industrial Automation Systems

In the context of developing a visual monitoring system for the glass bottle manufacturing industry, it is crucial to address the societal and global issues surrounding industrial automation systems while incorporating principles of sustainable development. This section explores the key considerations related to societal impact and sustainability, emphasizing the need to align the project with broader goals of societal well-being and environmental responsibility.

The integration of societal and global issues in the development of the visual monitoring system aims to ensure that the project contributes positively to the well-being of individuals, communities, and the environment. By examining the social, economic, and environmental implications, we can make informed decisions that foster sustainable development and mitigate potential negative impacts.

One of the key aspects of societal and global considerations is the impact on employment and labor dynamics. The implementation of an automated system should be mindful of its potential effects on the workforce and seek to provide adequate support and opportunities for workers affected by automation. Moreover, the system should promote equitable and inclusive employment practices, fostering job security and fair working conditions.

Environmental sustainability is another critical dimension that needs to be addressed. By optimizing energy consumption, reducing waste generation, and minimizing the system's ecological footprint, we can ensure that the visual monitoring system contributes to a more sustainable and environmentally responsible manufacturing process. This includes exploring energy-efficient technologies, adopting green practices, and implementing strategies for waste reduction and recycling.

Furthermore, the social acceptance and perception of automation technologies play a significant role in their successful implementation. Engaging with stakeholders and the local community throughout the development process can help address concerns, build trust, and ensure that the visual monitoring system is well-received and widely supported.

By incorporating societal and global considerations, along with sustainable development principles, into the development of the visual monitoring system, we aim to create a solution that goes beyond technical efficiency and actively contributes to societal well-being, environmental protection, and sustainable economic growth.

1.4 Project Objective

The main aim of this project is to present a efficient and effective methodology to develop a visual monitoring system to enhance the manufacturing process by integrating a user-friendly Human-Machine Interface (HMI) through Visual Basic (VB.NET) and employs seamless interaction with image processing techniques utilizing Convolutional Neural Networks (CNNs) as a deep learning algorithms for glass bottle monitoring.. Specifically, the objectives are as follows:

a) To design and develop a VB.NET-based visual monitoring system that is tailored to monitor crucial processes within the glass bottle manufacturing plant. This system will provide a user-friendly interface for effective control and monitoring of manufacturing operations.

b) To integrate camera sensor and image recognition technique using Convolutional Neural Networks (CNNs) algorithm for the visual monitoring system in the glass bottle manufacturing plant, reducing manual inspection and increasing efficiency.

c) To implement a system that capable of providing a real-time monitoring capabilities for the glass industry that can be integrated with existing industrial automation systems, enabling communication with equipments such as conveyor.

1.5 Scope of Project

This project will be developed within a specific scope, which sets the boundaries for the final system. The system will need to adhere to the following requirements:

a) Utilize one of the languages in Microsoft Visual Studio, VB.NET to create a simple and user-friendly interface for the visual monitoring system.

b) Incorporation of USB cameras as sensor hardware, strategically integrated with a physical conveyor system, replicating the glass bottle production line.

c) The system will only showcase the assembly conveyor for monitoring tasks present in the glass bottle production line.

d) Utilization of Python language supported by OpenCV and Keras for implementing Convolutional Neural (CNNs) algorithm as a deep learning method for image recognition in the defect detection system.

e) The developed model specialized in predicting whether a glass bottle is in proper or defective condition by leveraging a pre-trained dataset, with a specific emphasis on shape defects.

f) Detection of shape defects, specifically focusing on breakages in critical areas like the bottle neck and mouth.

g) The system includes functionality to recognizes bottle colours by analyzing the RGB values of the central pixel and matching them with a pre-existing dataset.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter aims to identify the factors that contribute to the dangerous nature of these processes where human intervention is risky and not recommended. It will also explore the selection and evaluation of suitable automated visual monitoring systems for the project and how they can effectively achieve the project's objectives. A thorough overview of current research and technologies is given in the literature study for the visual monitoring and inspection systems. By reviewing existing research and studies, this chapter will provide insights into the challenges and solutions in implementing visual monitoring systems for inspection processes in the glass bottle manufacturing industry.

2.2 Glass Bottle Manufacturing Industry

The glass bottle manufacturing industry plays a significant role in various sectors such as beverages, cosmetics, pharmaceuticals, and more. Glass bottles are preferred for their aesthetic appeal, durability, and ability to preserve the quality of the contents. Understanding the manufacturing process and the materials involved is crucial for ensuring efficient production and maintaining product integrity. This section presents a comprehensive literature review of the glass bottle manufacturing industry, discussing key aspects such as the manufacturing process, materials used, and the different stations and production lines involved.

The manufacturing process of glass bottles involves several distinct steps that contribute to the final product's quality and functionality. The process typically begins with the selection and preparation of raw materials, including silica sand, soda ash, limestone, and other additives. Silica sand, which consists primarily of silicon dioxide (SiO_2), is the main component and provides the glass with its structural strength. Soda ash (sodium carbonate) and limestone (calcium carbonate) act as fluxes, reducing the melting temperature of the batch and facilitating the fusion of the ingredients.[1]

The precise composition of the glass batch is crucial for achieving the desired glass quality and performance. Factors such as the particle size distribution of the raw materials, batch homogeneity, and the presence of impurities significantly impact the glass's characteristics. Once the raw materials are prepared, the batch is fed into a furnace for melting. The furnace, operating at temperatures around 1500°C, transforms the solid batch into a molten state.[1] The furnace design and operation play a crucial role in achieving consistent and homogeneous glass melt. Various types of furnaces are employed in the glass bottle manufacturing industry, such as regenerative, recuperative, and electric furnaces, each with their advantages and limitations. Maintaining the proper temperature and atmosphere inside the furnace is essential for achieving optimal glass quality and reducing energy consumption.

The molten glass is then shaped into bottles using various techniques, depending on the desired bottle design and production requirements. The most common shaping methods are blowing and pressing. In the blow-blow process, a parison (a hollow tube of molten glass) is formed by blowing compressed air into a mold, followed by the shaping of the parison using a final blow [2]. Press-blow, on the other hand, involves pressing the molten glass into a mold, followed by blowing compressed air to achieve the desired bottle shape.

Both techniques require precision and control to ensure uniform wall thickness, accurate dimensions, and consistent bottle quality.

In a glass bottle manufacturing plant, different stations and production lines are established to ensure a smooth and efficient process flow. These stations include the batch house, furnace, forming machines, annealing lehr, and inspection stations [2]. The batch house is responsible for handling and preparing the raw materials, ensuring the correct composition of the batch. It involves processes such as weighing, mixing, and refining the raw materials to achieve the desired quality. The batch is then transported to the furnace, where it is fed into the melting chamber.

The furnace, operating at high temperatures, plays a critical role in the glass bottle manufacturing process. It melts the batch into molten glass, facilitating the fusion of the raw materials. The furnace temperature and duration of the melting process are carefully controlled to ensure proper glass consistency and quality.

Once the molten glass is formed, it is directed to the forming machines, also known as the "hot end" of the production line. These machines shape molten glass into bottles based on predefined molds and techniques. The blow-blow and press-blow methods are widely employed for their versatility and efficiency. In the blow-blow process, a parison is formed by blowing compressed air into the molten glass, followed by shaping and final blow to achieve the desired bottle shape [2]. The press-blow technique involves pressing the molten glass into a mold, followed by blowing compressed air to refine the shape. Advanced computer-controlled systems ensure precise control of the forming process, resulting in consistent bottle dimensions and quality.

To ensure the strength and stability of the newly formed bottles, they undergo a process called annealing. The annealing lehr is a temperature-controlled chamber where the bottles are gradually cooled to relieve internal stresses within the glass [1]. This controlled cooling process, often accompanied by a carefully controlled cooling profile, allows the glass structure to relax and solidify in a controlled manner. Annealing prevents the bottles from cracking or breaking due to thermal stresses and enhances their overall strength and durability.

Throughout the production line, inspection stations are strategically placed to detect and eliminate any defects that may compromise the quality of the glass bottles. Visual inspection techniques are commonly employed to identify surface imperfections, such as scratches or bubbles. Automated systems equipped with machine vision technology have also gained prominence in defect detection. These systems utilize advanced algorithms to analyze bottle images and detect defects such as cracks, uneven thickness, or improper dimensions with high accuracy and speed [3]. The integration of automated inspection systems helps to ensure that only bottles of the highest quality proceed further in the production line.

In conclusion, the glass bottle manufacturing industry encompasses a complex and multi-step process that involves the selection and preparation of raw materials, high-temperature melting, shaping techniques, annealing, and quality control measures. The precise composition of the glass batch, furnace design, shaping methods, and annealing processes significantly influence the quality, strength, and aesthetic appeal of the glass bottles.

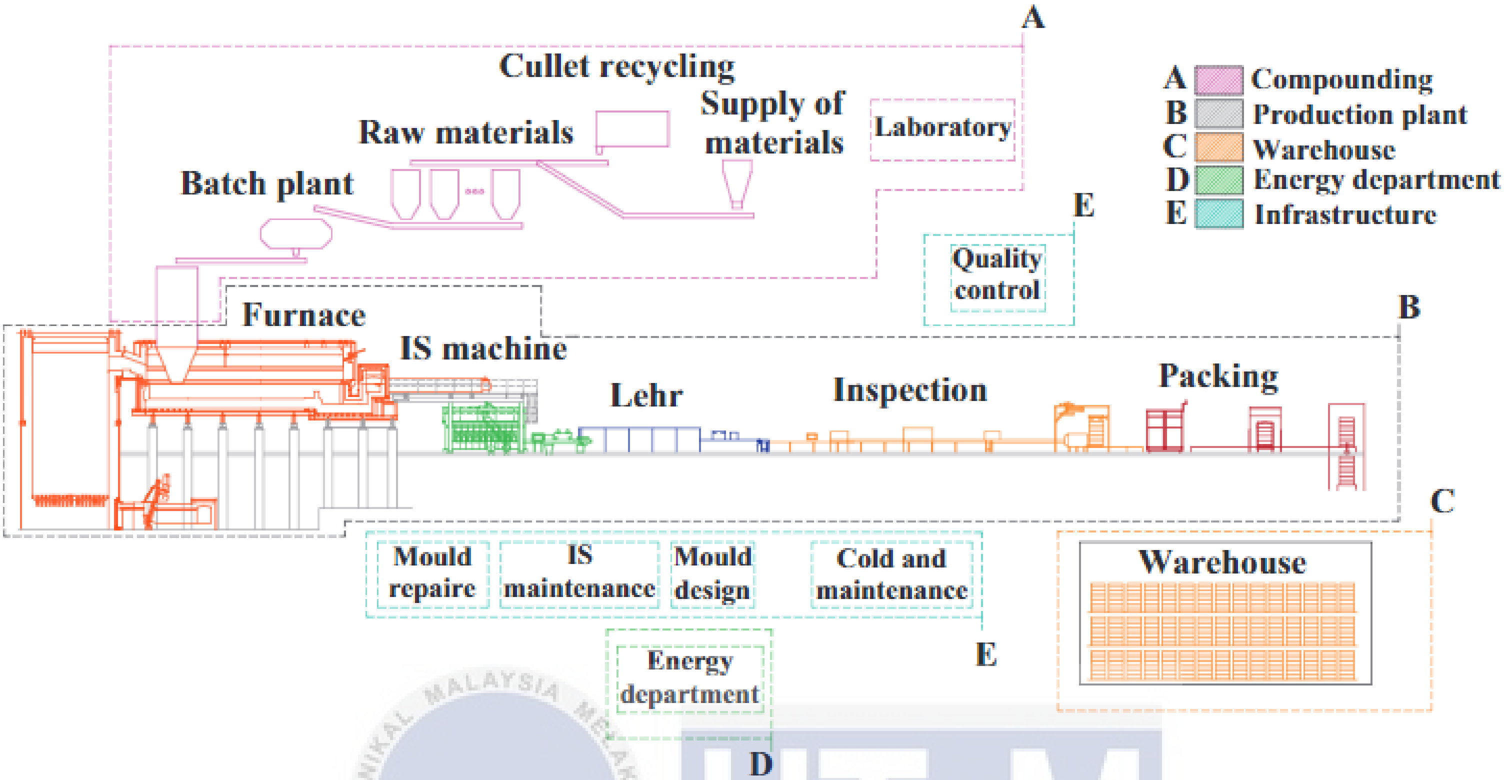


Figure 2-1 Technological Schema of the Production Procedure [2]

Table 2.1 Components and Functions of a Glass Bottle Manufacturing Production Line

Station	Function
Batch House	Raw material handling and preparation
Furnace	Melting the batch and transforming it into glass
Forming Section	Shaping the molten glass into bottles
Annealing Lehr	Controlled cooling to relieve internal stresses
Inspection Stations	Quality control and defect detection

2.3 Comparison with Metal and Rubber Manufacturing Industries

The glass bottle manufacturing industry shares both similarities and differences with other manufacturing industries such as the metal and rubber industries. By examining the similarities and differences between the glass bottle, metal, and rubber manufacturing industries, we can uncover valuable insights into the potential advantages of implementing a visual monitoring system. Such a system has the potential to enhance process efficiency, reduce waste, improve product quality, and ensure the safety of manufacturing operations. With a deeper understanding of how these industries align and diverge, we can identify the specific areas where a visual monitoring system can be most beneficial, leading to advancements in process control, defect detection, and overall productivity

2.3.1 Similarities in Manufacturing Processes

The glass bottle, metal, and rubber industries exhibit similarities in manufacturing processes. They involve raw material preparation, shaping and forming, heat treatment, and quality control. Raw materials are carefully selected, handled, and processed through refining, mixing, and compounding. Techniques like molding, extrusion, casting, and forging shape the materials into desired forms. Heat treatment enhances material properties through controlled heating and cooling. The annealing process in the glass bottle industry relieves stresses and strengthens bottles. Metals undergo heat treatment processes like annealing, tempering, or quenching. Rubber materials are improved through vulcanization. Stringent quality control measures are implemented in all three industries, utilizing inspection stations, testing equipment, and quality assurance protocols to ensure products meet standards.

.

2.3.2 Differences in Manufacturing Processes

The glass bottle, metal, and rubber industries differ in their manufacturing processes. Glass is inorganic, metals are metallic elements or alloys, and rubber is an organic polymer. Glass bottles are formed through blowing or pressing, while metal components are shaped through casting, forging, or machining. Rubber products are made using molding or extrusion techniques. Heating methods vary, with glass using high-temperature furnaces, metals requiring specific heating methods like induction or electric arc furnaces, and rubber undergoing vulcanization. Metal components undergo additional machining processes for shaping and finishing, while glass bottles and rubber products require minimal machining. Implementing a visual monitoring system benefits all three industries, improving quality control and safety through real-time monitoring and early defect detection. Visual monitoring technologies optimize operations and drive continuous improvement.

.

2.4 Hazardous and Critical Processes in Glass Bottle Manufactory

The glass bottle manufacturing process involves several critical and hazardous processes that pose risks to both worker safety and product quality. Understanding these processes and their associated hazards is crucial for implementing appropriate safety measures and mitigating potential risks. This section provides a comprehensive literature review of the critical and hazardous processes in glass bottle manufacturing, highlighting the specific risks involved and the strategies recommended to address them.

According to Seker, [4] the glass manufacturing process entails various hazards and risks. Similar to other manufacturing industries, the glass manufacturing sector is not immune to its own set of dangers that pose threats to individuals working in proximity to operating machinery and handling heavy glass materials. In line with this, [5] has identified six major hazards commonly encountered in glass manufacturing workplaces, which are detailed below:

I. Physical Injuries: Workers may be at risk of cuts, punctures, and lacerations from handling sharp-edged glass objects.

II. Thermal Hazards: The glass manufacturing process involves high temperatures, exposing workers to the risk of burns and heat-related stress.

III. Chemical Hazards: Exposure to hazardous substances, such as silica dust generated during glass production, can lead to respiratory problems and other health issues.

IV. Ergonomic Hazards: Poor ergonomics, including repetitive movements, awkward postures, and lifting heavy loads, can contribute to musculoskeletal disorders and injuries.

V. Mechanical Hazards: Moving parts of glass manufacturing machinery can pose risks of crushing, entanglement, and other mechanical injuries if safety precautions are not followed.

VI. Falls and Trips: Slippery surfaces, inadequate lighting, cluttered work areas, and uneven floors increase the risk of slips, trips, and falls, potentially causing injuries.

2.4.1 Batch House

The batch house is a critical station in glass bottle manufacturing, where raw materials are handled and prepared. It involves hazardous processes that require careful management. Raw material handling poses health risks, such as respiratory issues from silica dust [4]. Ventilation systems, PPE, and air quality monitoring protect workers. Weighing and mixing use equipment with moving parts that can cause injuries if not operated or maintained correctly. Fire and explosion hazards exist due to combustible materials. Chemical hazards arise from additives and processing chemicals, requiring proper labeling, storage, and training. Manual handling of heavy materials can cause musculoskeletal injuries without following proper lifting techniques.

2.4.2 Furnace

The furnace is a crucial station in glass bottle manufacturing, where raw materials are melted into molten glass [6]. It involves critical and hazardous processes that require careful management. Furnaces used in glass bottle manufacturing operate at extremely high temperatures, typically ranging from 1,400 to 1,600 degrees Celsius [7]. The high temperatures can cause severe burns and heat-related injuries to workers if proper safety measures are not in place. Furnace operation control prevents explosions or fires, requiring continuous monitoring and regular maintenance. Handling molten glass demands proper techniques and specialized tools to minimize burn risks. Fuels or energy sources are used to maintain high temperatures. Environmental and worker health concerns arise from potential emissions or by-products, requiring proper control measures. Furthermore, ensuring proper ventilation systems and implementing effective pollution control measures are essential to minimize the environmental impact of furnace operations and maintain a safe and healthy work environment.

2.4.3 Forming Machines

Forming machines are crucial in shaping molten glass into bottles, but they involve critical and hazardous processes that require attention to ensure worker safety and high-quality production. Mechanical injuries pose a significant risk due to moving parts, molds, and pneumatic systems. Accidents like crushed fingers or entanglement can occur from improper operation or mechanical failures [5]. Thermal injuries can result from direct contact with hot glass or surfaces. Handling and loading molds also require caution to prevent injuries. Additionally, defects like uneven thickness or bubbles can compromise bottle integrity. Real-time visual monitoring systems help identify and address defects, minimizing production of defective bottles.

2.4.4 Annealing Lehr

The annealing lehr in glass bottle manufacturing is crucial for strengthening bottles but also involves hazards and critical processes. One of the significant hazards associated with the annealing lehr is the risk of thermal injuries [8]. Moreover, the annealing lehr may also present challenges related to bottle inspection and quality control. During the cooling process, bottles may develop defects such as cracks, chips, or uneven cooling, which can compromise their integrity. Another critical aspect of the annealing lehr is the handling and movement of bottles during the cooling process. The bottles need to be carefully loaded onto and unloaded from the lehr conveyor system to avoid breakages or accidents.

2.4.5 Inspection and Quality Control

Inspection and quality control are critical stages to ensure that the glass bottles meet the required standards. Hazards in this process include the presence of sharp edges or fragments on the bottles, which can cause cuts or injuries if not handled carefully. Workers involved in inspection should be cautious and equipped with cut-resistant gloves and other appropriate personal protective equipment to minimize the risk of injuries. It is also essential to have proper lighting and ergonomic workstations to enhance visual inspection accuracy and reduce the risk of errors or missed defects.

2.4.6 Packaging and Transportation

The final critical process in glass bottle manufacturing is the packaging and transportation of the finished products. Hazards at this stage include the risk of dropping or mishandling the bottles, leading to breakage and potential injuries. Manual handling of heavy boxes or pallets can also result in musculoskeletal strains and injuries if proper lifting techniques are not followed.

Table 2.2 Hazardous and Critical Processes in Glass Bottle Manufactory

Station	Critical Process	Hazards
Raw Material Handling	Material intake and inspection	Physical injuries, inhalation of dust
Batch House	Mixing and weighing	Silica dust exposure, burns and injuries
Furnace and Melting	Melting	High temperatures, release of hazardous gases
Forming Machine	Glass shaping and forming	Mechanical hazards, burns
Annealing	Controlled cooling	Thermal hazards, glass breakage, risk of cuts
Inspection and	Visual inspection	Eye injuries, exposure to sharp

Station	Critical Process	Hazards
Quality Control		edges or defective bottles
Packaging	Packaging and labelling	Manual handling injuries, cuts and punctures
Shipping and Transportation	Loading and transportation	Manual handling injuries, breakage and damage during transit

In conclusion, the glass manufacturing industry involves critical and hazardous processes that require careful attention to ensure worker safety and the production of high-quality glass products. By integrating visual monitoring systems into the glass manufacturing process, companies can enhance workplace safety, reduce the likelihood of accidents, and improve overall productivity.

2.5 Visual Monitoring

Visual monitoring is a valuable tool that enables real-time observation and analysis of various processes and activities in manufacturing industries. It involves the use of cameras, sensors, and imaging technologies to capture and analyze visual information, providing insights into the production line, equipment performance, and product quality. By harnessing the power of visual monitoring, manufacturers can enhance process control, detect anomalies, and make informed decisions to optimize efficiency and ensure consistent product quality.[9]

Visual monitoring systems are designed to capture and process visual data in a continuous manner. They can be integrated into different stages of the manufacturing process, including raw material handling, forming, assembly, inspection, and packaging. These systems employ advanced image processing algorithms to analyze the captured images or video streams, enabling real-time monitoring and analysis of critical parameters.

The benefits of visual monitoring in manufacturing industries are numerous. Firstly, it provides operators and supervisors with real-time visibility into the production line, allowing them to monitor process parameters, identify bottlenecks, and make necessary adjustments promptly [10]. This proactive approach helps in optimizing production efficiency, reducing downtime, and minimizing wastage.

Secondly, visual monitoring systems enable early detection of abnormalities or deviations from standard operating conditions. By continuously analyzing the visual data, these systems can identify potential defects, equipment malfunctions, or process deviations [11]. Timely detection allows for prompt intervention, preventing further production of defective products and reducing scrap rates. Moreover, it helps in identifying root causes and implementing corrective actions to address underlying issues, ensuring consistent product quality.

Furthermore, visual monitoring can assist in the implementation of preventive maintenance strategies. By monitoring equipment performance and capturing visual information, potential equipment failures or abnormalities can be identified in advance [10]. This enables scheduled maintenance activities, minimizing unplanned downtime and optimizing the overall equipment effectiveness.

Additionally, visual monitoring systems facilitate data-driven decision making. By capturing and analyzing visual data, manufacturers can extract valuable insights and trends related to process performance, product quality, and equipment efficiency. These insights can be used to optimize process parameters, improve product design, and enhance overall manufacturing operations.

It is important to note that visual monitoring systems should be designed and implemented in accordance with industry standards and best practices. Considerations such as camera placement, lighting conditions, image resolution, and data storage and analysis capabilities should be considered to ensure accurate and reliable monitoring. [12]

In conclusion, visual monitoring plays a vital role in manufacturing industries by providing real-time visibility, detecting abnormalities, optimizing process control, and facilitating data-driven decision making. As technology continues to advance, visual monitoring systems are expected to become more sophisticated, offering enhanced capabilities and further contributing to the efficiency, quality, and safety of manufacturing processes.

2.5.1 Importance of Visual Monitoring in the Glass Bottle Manufactory

Visual monitoring plays a critical role in the glass bottle manufacturing industry, ensuring product quality, process efficiency, and safety throughout the production processes. By employing advanced visual inspection techniques, operators can visually assess different stages of bottle manufacturing, identify defects, and ensure adherence to quality standards.

The importance of visual monitoring in the glass bottle manufacturing industry has been widely recognized. Research studies have highlighted its significance in ensuring product quality and minimizing defects. For example, Salman Shaikat et al. conducted a study focused on visual monitoring in glass bottle production, emphasizing its role in detecting defects such as cracks, impurities, or uneven glass distribution [13]. By promptly identifying these issues, manufacturers can take corrective actions to maintain high-quality standards and avoid product rejections or recalls.

Visual monitoring also plays a crucial role in ensuring the safety of the manufacturing processes in the glass bottle industry. Operators can visually inspect the machinery, equipment, and production lines to identify potential hazards or malfunctions that may lead to accidents or injuries. This allows for timely intervention and implementation of safety measures. Kumar et al conducted a study on visual monitoring systems in the glass manufacturing industry, highlighting its effectiveness in improving workplace safety and reducing the occurrence of hazardous incidents [14].

In addition to quality control and safety, visual monitoring contributes to process efficiency in the glass bottle manufacturing industry. Operators can visually monitor various production parameters, such as glass distribution, mold alignment, or filling accuracy, to identify areas for process optimization and waste reduction [3]. Real-time feedback obtained through visual monitoring enables operators to make data-driven decisions and implement corrective measures, ultimately improving production efficiency.

The integration of visual monitoring systems in the glass bottle manufacturing industry has proven to be highly beneficial. These systems, equipped with cameras, sensors, and computer vision algorithms, provide real-time visual data, enabling operators to detect defects, ensure product consistency, and maintain high-quality standards [15]. They also contribute to the optimization of production processes, reduction of waste, and enhancement of overall operational efficiency.

In conclusion, visual monitoring is of utmost importance in the glass bottle manufacturing industry. It ensures product quality, enhances process efficiency, and improves workplace safety. By leveraging advanced visual inspection techniques and monitoring systems, manufacturers can achieve high-quality standards, reduce defects, and optimize their manufacturing processes.

2.5.1.1 Ensuring Safety and Hazard Control in the Glass Bottle Manufactory

Visual monitoring is essential for ensuring safety in the glass bottle manufacturing industry. It plays a vital role in identifying potential hazards and controlling safety risks in specific manufacturing processes.

During the glass melting and forming processes, visual monitoring enables operators to monitor critical aspects such as temperature, viscosity, and flow of molten glass, ensuring safe handling and reducing the risk of accidents. By visually inspecting furnace operations, glass distribution, and mold alignment, operators can promptly detect equipment malfunctions, glass breakage, or overheating, allowing for immediate corrective actions to prevent accidents, injuries, and equipment damage.

The integration of visual monitoring systems with advanced technologies, such as computer vision and artificial intelligence, further enhances safety practices in the industry [14]. These systems can be programmed to detect safety-related anomalies or deviations, such as the presence of foreign objects or abnormal glass flow, providing real-time alerts to operators for swift intervention.

In conclusion, visual monitoring in the glass bottle manufacturing industry contributes to safety by enabling operators to monitor critical processes, identify potential hazards, and take immediate corrective actions. It is particularly beneficial in the glass melting and forming processes, where the handling of molten glass and equipment operations present inherent risks.

2.5.2 Existing Visual Monitoring System in Manufacturing Industry

In today's rapidly evolving manufacturing industry, ensuring product quality, process efficiency, and worker safety are critical factors for success. Visual monitoring systems have emerged as indispensable tools for achieving these goals. These systems leverage advanced imaging technologies and computer vision algorithms to capture, analyze, and interpret visual data in real-time. By providing valuable insights into production processes, visual monitoring systems help manufacturers identify defects, streamline operations, and improve overall productivity.

This literature review aims to explore the existing visual monitoring systems utilized in the manufacturing industry. It provides an overview of the key technologies, applications, benefits, and challenges associated with these systems. By examining the current state-of-the-art in visual monitoring, this review aims to identify gaps in research and highlight potential avenues for future development and improvement.

2.5.2.1 Machine Vision Systems

Machine vision technology has a rapid development rate in recent year especially in intelligent manufacturing [16]. Machine vision systems have become integral in the manufacturing industry for automated inspection and quality control processes. These systems utilize advanced imaging technology, image processing algorithms, and artificial intelligence to analyze visual data and make objective decisions regarding product quality. In recent years, there has been a significant amount of research and development in the field of machine vision systems, leading to advancements in hardware, software, and algorithms. Machine vision systems can perform a wide range of inspection tasks, such as dimensional measurement, defect detection, surface inspection, and barcode reading [17]. The ability to automate these tasks reduces reliance on manual inspections, decreases human error, and improves overall productivity.

2.5.2.2 Camera-based Monitoring Systems

Camera-based monitoring systems have become an integral part of the manufacturing industry, providing real-time visual information for process monitoring, quality control, and safety assurance [18]. These systems utilize high-resolution smart cameras strategically placed throughout the manufacturing facility to capture video footage of the production processes [19]. The use of camera-based monitoring systems offers several advantages in the manufacturing industry. Firstly, they provide real-time visibility into the production processes, allowing operators to monitor operations remotely and make informed decisions promptly. By continuously monitoring the processes, deviations or anomalies can be detected early, enabling timely intervention to prevent quality issues, equipment failures, or safety hazards.

2.5.2.3 Infrared Thermography Systems

Infrared thermography systems use thermal imaging cameras to capture and analyze the heat patterns emitted by objects [20]. These systems are especially useful for monitoring temperature variations, identifying hotspots, and detecting equipment malfunctions. In manufacturing, infrared thermography systems are employed for predictive maintenance, energy efficiency assessment, and process optimization. They can detect issues like overheating, insulation problems, or inadequate heat transfer, enabling timely interventions to prevent equipment failures or improve energy utilization.

2.5.2.4 Sensor-based Monitoring Systems

Sensor-based monitoring systems are widely used in the manufacturing industry to gather real-time data on various parameters and monitor critical processes [21]. Sensor-based monitoring systems integrate visual data with data from other sensors, such as temperature, pressure, or vibration sensors [22]. These systems provide a holistic view of the manufacturing process, allowing operators to monitor multiple parameters simultaneously. By combining visual information with other sensor data, operators can gain deeper insights into process conditions, equipment performance, and product quality. Sensor-based monitoring systems find applications in industries such as chemical processing, oil and gas, metal, rubber, and automotive manufacturing.

2.5.2.5 Supervisory Control and Data Acquisition (SCADA) Systems

Supervisory Control and Data Acquisition (SCADA) systems play a crucial role in the manufacturing industry by providing real-time monitoring, control, and data acquisition capabilities. These systems utilize a combination of hardware and software components to gather data from various sensors and devices, enabling operators to visualize and analyze critical information about the manufacturing processes [23]. SCADA systems enable remote monitoring and control, providing operators with the flexibility to oversee manufacturing processes from a control room or even from off-site locations. This remote accessibility enhances operational efficiency, reduces the need for constant physical presence, and improves safety by minimizing exposure to hazardous environments.

2.5.2.6 Colour Inspection Systems

Colour inspection systems are widely used in the manufacturing industry to ensure accurate colour representation and consistency in products. These systems utilize advanced imaging technology and colour sensors to analyze the colour properties of objects or surfaces. By comparing the measured colour values against predefined standards or reference samples, colour inspection systems can detect colour variations, defects, or inconsistencies that may occur during the manufacturing process [24]. These systems are particularly valuable in industries where colour accuracy is crucial, such as automotive, textiles, printing, and cosmetics. By detecting and addressing colour-related issues in real-time, colour inspection systems help manufacturers maintain product quality. With their ability to provide objective and reliable colour analysis, colour inspection systems are essential tools for ensuring consistent and visually appealing products in the manufacturing industry.

2.6 Recent Development of Visual Monitoring and Inspection System in Industrial Automation (Glass Bottle Industry)

Pinhole camera model and lens distortion model of camera estimation is considerably used [25]. Machine vision- predicated canend examination system has been mooted by Chen etal. [26]. Saliency discovery and template matching ways have been used for visual examination of the bottom of the glass bottle [27]. Recently deep knowledge predicated fashion is used to classify bottles using machine vision approach [28][29].

Huang et al. [30]. has presented a vision-based system for empty bottle inspection. Key technologies in empty bottle inspection systems are studied to solve detecting error and poor adaptability problems. Those technologies have two different approaches: the ones in the first grouplocate and track bottle mouth, bottle bottom and walls while the other group technologies involve defect detecting. The article proposes distinctive algo-rithms for bottle locating, tracking and defect detecting based on inspection requirements and images of bottle mouth, bottom and walls.

Gong et al. [31]. present a comprehensive study on a machine vision system designed for the online inspection of transparent label defects on curved glass bottles. The authors employ an area-array camera in conjunction with a specially crafted blue dome illumination device to capture high-quality standstill images, effectively mitigating issues related to reflections. A notable contribution of their work lies in the introduction of deformable template matching, a technique tailored for accurate defect localization on the inherently curved surfaces of glass bottles. The defect detection process is further optimized through an adaptive threshold selection strategy that combines global and local threshold values, incorporating a Gaussian fitting algorithm to enhance the identification of small scratches. The system also utilizes skeleton extraction and distance transformation techniques, specifically tailored for the identification of printing errors on the golden edge of the transparent labels. Remarkably, the proposed system attains an impressive detection accuracy of 99.5% while operating at a speed of 60 bottles per minute, showcasing its efficacy in significantly improving quality control processes within the glass bottle manufacturing industry.

Salman Shaikat et al., n.d. [13]. introduce an intelligent system tailored for the detection of bubble and crack defects in glass bottles through the application of image processing techniques, particularly the Haar Cascade method. The system is equipped with a camera and a stepper motor, enabling a 360-degree rotation of the glass bottle for a comprehensive inspection process. Upon the identification of a defect, the system promptly captures and stores an image for subsequent in-depth analysis. This innovative approach not only showcases the adaptability and precision of the Haar Cascade method but also underscores the importance of comprehensive examination through the bottle's complete rotation. The efficacy of the proposed system is substantiated through collected data, affirming its potential as a highly efficient application for the detection of defects in glass bottles. By contributing to the field of automated quality control systems in the glass manufacturing industry, this research signifies a notable step forward in ensuring heightened product quality and reliability in the production of glass bottles.

Kumar at el. [14]. propose a comprehensive design and implementation of a remote diesel level monitoring system integrated with an automated fire extinguishing system utilizing wireless sensor networks. The primary objective is to sense and wirelessly transmit diesel levels using Zigbee technology to a remote base station for processing and analysis. Given the high inflammability of diesel in generators, the authors address safety concerns by incorporating an automated fire extinguishing system, enhancing the overall security of the setup. To further fortify the safety measures, the system is designed to send

SMS alerts through a GSM system, ensuring timely notifications to officials in case of anomalies. Notably, the system's adaptability to multilingual alerts caters to the diverse linguistic environment in industries. This innovative solution specifically targets the monitoring of diesel levels in the context of glass manufacturing industries, where the integration of an automated fire extinguishing system adds a layer of safety and efficiency to the production processes.

Li at el. [32] presented a method for detecting glass bottles based on the connectivity domain feature, aiming to enhance both the quality and detection efficiency of glass bottles.This method extracts the defect features by pre-processing the collected image and threshold segmenting the image of the bottle. The method of analyzing the aperture area and width of the connected area of the bottle mouth is used to detect the image of the bottle mouth. The matching of the connected domain pixels is used to judge whether the bottle mouth is qualified and detect the defect location, and use the range of the detected defects to be compared with the range of the manually calibrated defects. The experimental results of bottle detection show that this method can accurately determine whether the bottle mouth is defective and detect the defect range of the bottle. The detection rate has reached a high level with good detection accuracy.

In their work, *X. Zhou et al.* [27] introduced a machine vision-based apparatus designed for real-time inspection of bottle bottoms, with a particular emphasis on developing an efficient defect detection framework. The methodology integrates saliency detection and template matching techniques to comprehensively analyze and evaluate the quality of the bottle bottoms. The initial step involves locating the bottom of the bottle through Hough circle detection and size prior considerations. Subsequently, the region of interest is strategically divided into three measurement regions to facilitate a detailed inspection process. Saliency detection is applied to the central panel region, while multiscale filtering is employed for the annular panel region. For the annular texture region, a combination of template matching and multiscale filtering is utilized. The results of defect detection from these diverse techniques are fused to provide a holistic assessment of the bottle bottom quality. The efficacy of the proposed framework is thoroughly evaluated using images obtained from the designed apparatus, showcasing its potential as a real-time and robust solution for inspecting bottle bottoms with a focus on defect detection.

In their paper, "Glass Bottle Bottom Inspection Based on Image Processing and Deep Learning," Koodtalang et al. [33] present an innovative system for inspecting glass bottle bottoms using a combination of image processing and deep learning techniques. The method begins with image processing to locate the bottle bottom, employing a median filter and high pass filter to eliminate noise. The Hough circle transform is then applied to detect the region of interest (ROI) in the bottle bottom. Subsequently, a cropped image is generated, and a square masked ROI image is obtained by masking unnecessary regions. This masked image is resized and input into a pre-trained predictive model, constructed using a deep convolutional neural network (CNN) with three convolutional layers and two fully connected layers. The entire system is programmed using Python, supported by OpenCV and Keras. Experimental results indicate impressive accuracies of 99.00% for bottom location and 98.50% for defect detection. The bottom location process takes only 22ms, while classifying defective bottles requires 48ms, demonstrating the system's real-time inspection capability. The proposed model not only achieves high accuracy but also offers efficiency in glass bottle inspection, showcasing its potential for practical applications in manufacturing quality control processes.

The article by Fu et al. [34] titled "Medicine Glass Bottle Defect Detection Based on Machine Vision," proposed a machine vision method for efficiently detecting defects in glass bottle ports, aiming for simplicity in operation and broad applicability. The approach involves acquiring bottle images using backlight illumination and subsequently preprocessing these images to assess glass bottle quality and determine defect ranges. Key preprocessing steps encompass image median filtering, image enhancement, and edge detection. The method is particularly adept at identifying various glass bottle defects, such as cracks, missing edges, dirty bottles, black spots, among others. Following extensive testing on medicinal glass bottles, the proposed method proves to be widely applicable to common glass bottle port defects. Notably, the study reports a recognition rate of 91.6% for thirty-six sample bottles exhibiting missing edge defects, showcasing the method's effectiveness in detecting and analyzing defects in medicine glass bottles. The simplicity of operation and versatility in addressing diverse defects suggest the potential practicality of this machine vision-based approach for quality control in pharmaceutical packaging.

In their study, Lu et al. [35] tackle the intricate task of quality inspection within the glass product industry, with a specific focus on detecting defects in glass bottle mouths. Despite the acknowledged potential of machine vision in surpassing manual inspection methods, the identification of screw thread defects presents a persistent challenge. To surmount this obstacle, the authors propose a novel bottle mouth detection method grounded in area segmentation. The methodology initiates with an area segment approach that employs traditional image processing methods, capitalizing on the unique characteristics of screw threads. The segmented bottle area is then partitioned into screw thread and non-screw thread regions. For the former, a defect detection method incorporating edge detection and Gaussian filters is introduced, precisely honing in on screw thread defects. In contrast, the non-screw thread region undergoes defect detection utilizing techniques such as the Sobel algorithm and global binarization to identify other types of defects. The effectiveness of the proposed method is thoroughly assessed using datasets obtained from a purpose-designed vision system. Experimental results unequivocally demonstrate the robust performance of the framework, showcasing its efficacy in achieving accurate defect detection. This research contributes significantly to the advancement of quality control processes in the glass product industry, highlighting the promising potential of the proposed area segmentation-based method for enhancing the visual inspection of glass bottle mouths.

2.6.1 Critical Analysis of Relevant Studies

Table 2.3 Critical Analysis of Relevant Studies

Study	Methodology and Techniques	Key Findings and Contributions	Limitations/Drawbacks
Huang et al. [30]	Vision-based system for empty bottle inspection	Distinctive algorithms for bottle locating, tracking, and defect detection	Limited evaluation of system performance
Gong et al. [31]	Machine vision system for online inspection of transparent label defects on curved glass bottles	Deformable template matching, adaptive threshold selection, skeleton extraction techniques	Limited discussion on computational efficiency
Salman Shaikat et al. [13]	Intelligent system for detecting bubble and crack defects in glass bottles	Haar Cascade method, comprehensive inspection through bottle rotation, high efficiency in defect detection	Limited validation with real-world production environments
Li et al. [32]	Method for detecting	Accurate detection of bottle	Limited analysis of performance

Study	Methodology and Techniques	Key Findings and Contributions	Limitations/Drawbacks
	glass bottles based on connectivity domain feature	mouth defects, comparison with manually calibrated defects	with different bottle types
X. Zhou et al. [27]	Machine vision-based apparatus for real-time inspection of bottle bottoms	Defect detection framework combining saliency detection and template matching techniques, quality assessment of bottle bottoms	Limited discussion on system scalability and real-time performance
Koodtalang et al. [33]	Glass bottle bottom inspection based on image processing and deep learning; Pre-processing, Hough circle transform, masking, and deep convolutional neural network for defect detection	High accuracy in bottom location (99.00%) and defect detection (98.50%) with real-time inspection ability; Efficient use of image processing and deep learning.	Challenges in real-world use, like scalability or adapting to different lighting conditions, are not explicitly discussed.
Fu et al. [34]	Medicine glass bottle defect detection based on image processing; Haar Cascade method,	Adaptability and precision of Haar Cascade method; Effective detection of various defects in medicinal glass bottles	Challenges linked to implementing the system into large-scale pharmaceutical production environments are not

Study	Methodology and Techniques	Key Findings and Contributions	Limitations/Drawbacks
	backlight illumination, image median filtering, edge detection		explicitly discussed.
Lu et al. [35]	Defect detection in glass bottle mouths based on area segmentation; Traditional image processing methods for screw thread defects	Robust performance in accurate defect detection; Potential improvement in visual inspection of glass bottle mouths	The study may not fully address how well the proposed method works for bottles with different shapes, sizes, or manufacturing conditions

2.7 VB.NET

VB.NET is a widely used object-oriented programming language that is part of the Microsoft .NET framework. It offers a range of features and tools that simplify the development of Windows-based applications. With VB.NET, developers can take advantage of modern programming concepts while still benefiting from the familiarity of earlier versions of Visual Basic. [36]

One of the key advantages of VB.NET is its ability to incorporate important programming concepts such as inheritance, polymorphism, and multithreading. This allows developers to build complex and scalable applications. Additionally, VB.NET provides access to the extensive class library of the .NET framework, offering a wide range of pre-built functionalities and components.

VB.NET is particularly well-suited for the development of user-friendly Windows applications. It provides a rich set of tools, controls, and libraries that facilitate the creation of professional-grade interfaces and user experiences. This, in turn, helps to improve user engagement and satisfaction.

Another strength of VB.NET lies in its database connectivity capabilities. It offers seamless integration with various database systems through ADO.NET, simplifying tasks such as data retrieval, manipulation, and storage. This makes it an ideal choice for developing data-driven applications that require efficient and secure data management.

VB.NET seamlessly integrates with other .NET languages, such as C# and F#, enabling developers to leverage existing code libraries and collaborate effectively within multi-language projects. This interoperability ensures that developers can make the most of their resources and expertise, promoting efficient and collaborative development processes.

To support VB.NET development, Microsoft provides Visual Studio as a comprehensive integrated development environment (IDE). Visual Studio offers a wide range of tools, debuggers, and project management features that enhance productivity and streamline the development workflow. It provides a user-friendly and intuitive interface that empowers developers to create high-quality applications efficiently.

Visual Basic .NET is settling into its new place as a top five programming language in the TIOBE index, which is based on search engine data and assesses popularity. VB.NET makes a good showing in the February 2019 report, after reaching a new high in the index. Indeed, among the top 20 languages tracked by TIOBE, VB.NET grew the most from the previous year's ranking, with a 3.02 percent gain. VB.NET is a GUI-based development tool that is faster than most other programming languages when it comes to

Rapid Application Development (RAD). VB also has a more basic syntax than other languages, a user-friendly visual environment, and excellent database connection. The differences between VB.NET and C# are listed below:

Table 2.4 Difference between VB.NET and C#

Feature	VB.NET	C#
Syntax	Uses a more English-like syntax	Uses a C-style syntax
Readability	Generally considered more readable for beginners	Requires a steeper learning curve
Variable Declaration	Variable are declared using keywords such as Private, Protected, Friend and Static, etc.	Variable are declared using declarations.
Object Creations	The object is created using New and Create Object().	The object can be created using New.
Object Initialization	Sun New() is used to initialize, use a newly created object.	In this, constructions are used to initialize the object.
Class Declaration	In VB.NET declare a class by using **Class <implementation>** keyword.	In C# declare a class by using the **Class** keyword.
Overloading Functions	In VB.NET for Overload a function or method **Overloads** keyword is used.	In C# for Overload a function or method no language keyword is required for this purpose.
Exponential Operator	It uses the ‘this’ operator.	It does not use the ‘this’ operator.

Feature	VB.NET	C#
Base Class	In VB.NET refer to a base class by using the **MyBase** keyword.	In C# refer to a base class by using the **base** keyword.
XML Literals	Supports XML literals for easy XML manipulation	Does not have built-in XML literals
Error Handling	Uses "On Error" statement for error handling	Uses "try-catch" blocks for error handling
Named Arguments	Supports named arguments in method calls	Supports named arguments in method calls (C# 4.0 and later)
Async Programming	Uses the "Async" and "Await" keywords for asynchronous programming	Uses the "Async" and "Await" keywords for asynchronous programming

2.8 Convolutional Neural Networks (CNNs)

One of the most widely used deep learning algorithms is the convolutional neural network (CNN). CNN was first introduced in the 1960s [37]. and has shown promising performance results in computer vision [38]. Developed to mimic the visual processing of the human brain, CNNs have become a cornerstone in various applications, ranging from image classification and object detection to medical image analysis and autonomous vehicles. At the core of CNNs is the concept of convolution, inspired by the human visual system's ability to recognize patterns through receptive fields. Unlike traditional neural networks, CNNs consist of layers with learnable filters, allowing them to automatically and adaptively learn hierarchical representations of data. This hierarchical feature extraction makes CNNs well-suited for tasks where the spatial arrangement of features is crucial, such as recognizing objects in images. The architecture of a typical CNN comprises convolutional layers, pooling layers, and fully connected layers [39]. Convolutional layers use filters to convolve across input data, capturing spatial hierarchies [40]. Pooling layers reduce dimensionality, retaining essential information. Fully connected layers connect all neurons, enabling classification based on learned features. CNNs have demonstrated remarkable success in image recognition challenges, notably with the advent of well-known architectures like AlexNet, VGG, GoogLeNet, and ResNet [41]. Transfer learning, leveraging pre-trained CNNs on large datasets, has further boosted their effectiveness across diverse domains.

2.8.1 General Model of CNNs

A typical Artificial Neural Network (ANN) has one input and output layer, plus multiple hidden layers. Each neuron takes an input vector X and produces an output Y using a function F, represented by the general equation : $Y=F(W \cdot X)$. [42]. Here, W is the weight vector, showing the strength of connections between neurons in adjacent layers. This weight vector is crucial for image classification. While pixel-based methods are common, considering the image's shape, or contextual information, often produces better results.

Convolutional Neural Networks (CNNs) are popular for contextual-based classification. A CNN consists of four components: (a) convolution layer, (b) pooling layer, (c) activation function, and (d) fully connected layer.

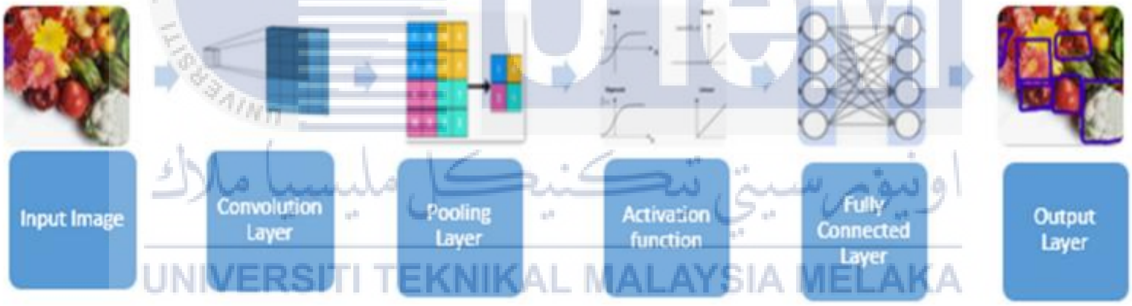


Figure 2-2 CNN Components [43]

2.8.1.1 Convolutional Layer

The Convolutional Layer is a fundamental component of Convolutional Neural Networks (CNNs). It plays a crucial role in capturing local patterns and spatial hierarchies within input data, making it particularly effective for image-related tasks. The layer involves the application of convolutional operations using learnable filters. These filters, also known as kernels, [43] slide across the input data, computing dot products at each position. The result is a feature map that highlights the presence of specific patterns or features.

The convolutional layer is characterized by parameters such as filter size (kernel size), the number of filters, padding, and stride. The filter size determines the spatial extent of the patterns the layer can capture. Padding is introduced to ensure that the filter covers the entire input, while the stride defines the step size of the filter movement. The output volume dimensions are calculated using the formula:

$$\text{Output Width} = \frac{\text{Input Width} - \text{Filter Width} + 2 \times \text{Padding}}{\text{Stride}} + 1$$

This process is instrumental in creating hierarchical representations of the input data, enabling the network to learn complex features.

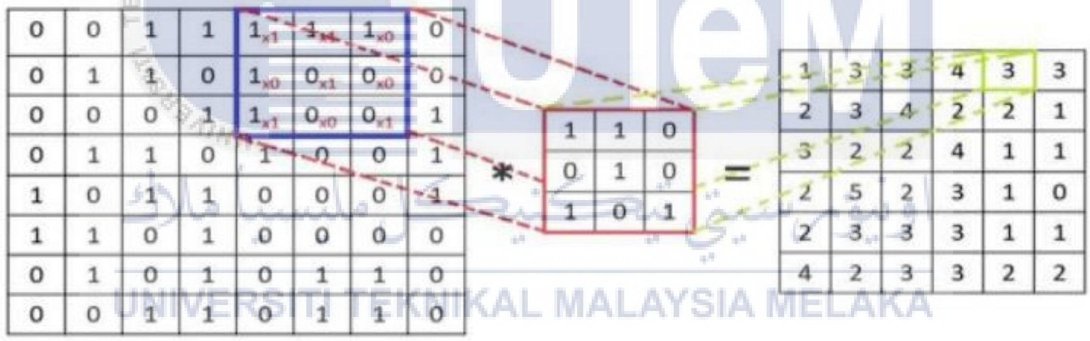


Figure 2-3 Convolutional Layer [44]

2.8.1.2 Pooling Layer

Pooling Layers are responsible for down-sampling the spatial dimensions of the input data, enhancing computational efficiency and extracting essential information. Max Pooling and Average Pooling are commonly used techniques within this layer. Max Pooling retains the maximum value within a defined window, effectively reducing the input size. If a pooling layer has a window size of 2×2 with a stride of 2, the output dimensions are halved in width and height. Average Pooling calculates the average value in the window, achieving a similar down-sampling effect. The pooling layer contributes to the network's translational invariance and reduces the risk of overfitting by focusing on the most salient features. It acts as a feature selector, retaining critical information while discarding less relevant details.

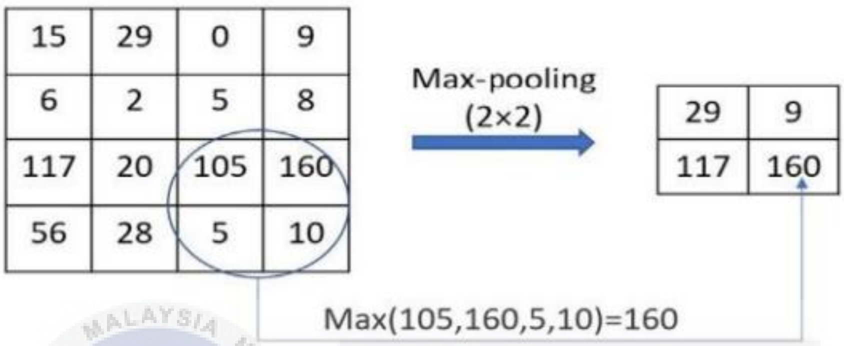


Figure 2-4 Pooling Layer [44]

2.8.1.3 Actication Function

Activation Functions introduce non-linearities to the neural network, allowing it to learn and represent complex relationships in the data. The ReLU (Rectified Linear Unit) activation function is widely used in CNNs [45]. It replaces negative values with zero, introducing non-linearity and enabling the network to efficiently learn intricate patterns. Other activation functions include Sigmoid and Tanh. The Sigmoid function squashes values to the range (0, 1), making it suitable for binary classification tasks. Tanh, on the other hand, maps values to the range (-1, 1) and is often used when the output should be centered around zero. Activation functions play a critical role in introducing non-linearity, enabling the network to approximate complex functions effectively.

2.8.1.4 Fully Connected Layer

Fully Connected Layers form the latter part of a CNN, where every neuron from one layer is connected to every neuron in the next. The input data is flattened into a vector, and weights are learned for each connection during the training process. If the input size is denoted as m and the output size as n, the number of parameters p is given by $p=(m+1)\times n$, considering the bias term.

These layers contribute to the network's ability to make predictions based on the learned hierarchical representations. Fully connected layers integrate high-level features extracted by previous layers, allowing the network to understand complex relationships in the data. They are crucial for tasks like image classification and object detection, where the network needs to make decisions based on the learned features. Each neuron in a fully connected layer receives input from all neurons in the previous layer, forming a dense connection structure [43]. This dense connectivity allows the network to model intricate dependencies and relationships, providing the necessary flexibility for accurate predictions. These layers serve as the final stages of feature extraction, enabling the network to transform the learned representations into meaningful predictions or classifications. They play a pivotal role in the overall effectiveness of CNNs for various tasks.

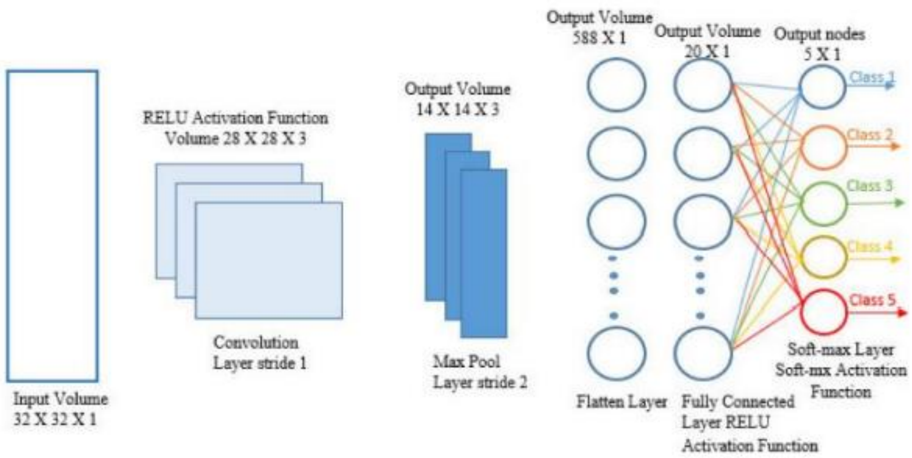


Figure 2-5 Fully Connected Layer [44]

2.9 Keras Library

Keras is a high-level neural network application programming interface (API) known for its user-friendly interface and versatility in developing, training, and deploying deep learning models . Offering an intuitive design, Keras caters to users of varying expertise, prioritizing simplicity and efficiency in neural network development. Its modular architecture enables the construction of complex models by assembling individual components like layers, optimizers, and activation functions, fostering flexibility and experimentation [46]. Compatible with popular backends such as TensorFlow, Theano, and Microsoft Cognitive Toolkit, Keras provides users the flexibility to choose the backend that aligns with project requirements. Furthermore, Keras supports high extensibility, allowing the creation of custom components, enhancing innovation in model development. The framework provides abstractions like the Sequential and Functional APIs, offering options for both straightforward and intricate model architectures. Integration with tools like TensorBoard facilitates model visualization and debugging, providing insights into model behavior during training. With a strong community, comprehensive documentation, and integration with TensorFlow, [47] Keras emerges as a preferred choice for diverse deep learning applications, combining power, accessibility, and extensive support resources.

2.10 OpenCV Library

OpenCV, or Open Source Computer Vision Library, is a dynamic software tool catering to computer vision and machine learning needs [48]. Compatible with languages like Python, Java, and MATLAB, OpenCV, written in C++, offers a broad suite of operations for image and video processing, object identification, and feature extraction. Specializing in real-time computer vision, it provides functionalities like camera calibration, image capture, and post-processing techniques. Noteworthy for its versatility, OpenCV supports deep learning integration with popular frameworks like TensorFlow and PyTorch, enabling developers to merge deep learning capabilities with computer vision expertise . Widely used in robotics, augmented reality, surveillance, healthcare, and automotive sectors, OpenCV's cross-platform compatibility and robust feature set make it a favored choice. With ongoing community-driven development, OpenCV remains a continually evolving platform for researchers and developers in the field of computer vision.

2.11 Canny Edge Detection

Canny edge detection, a seminal algorithm in computer vision, was introduced by John F. Canny in 1986 [49] and remains pivotal in image processing. Revered for its effectiveness, this technique excels in identifying edges with minimal noise and high accuracy. The multi-step process involves gradient calculation to pinpoint potential edges, non-maximum suppression for result refinement, and hysteresis-based edge tracking for comprehensive delineation [50]. The algorithm's widespread adoption extends to diverse applications, including object recognition, image segmentation, and feature extraction. Its precision and adaptability make it a fundamental tool for computer vision professionals, enhancing the accuracy of edge detection in various scenarios. As a go-to method, Canny edge detection significantly contributes to the success of applications requiring robust and nuanced edge detection capabilities.

2.12 Summary

In summary, the literature review presents a detailed exploration of the glass bottle manufacturing industry, emphasizing hazardous processes and the critical need for visual monitoring. A comparative analysis with metal and rubber manufacturing industries highlights both similarities and differences in manufacturing processes. The chapter underscores the importance of visual monitoring in ensuring safety and hazard control within the glass bottle manufacturing process. Existing visual monitoring systems, including Machine Vision, Camera-based Monitoring, Infrared Thermography, Sensor-based Monitoring, SCADA Systems, and Colour Inspection, are comprehensively reviewed. Recent developments in visual monitoring and inspection systems specific to the glass bottle industry are critically examined. The chapter then introduces the VB.NET programming language and provides an in-depth understanding of Convolutional Neural Networks (CNNs), covering their general model and essential components like Convolutional Layer, Pooling Layer, Activation Function, and Fully Connected Layer. The significance of the Keras library, OpenCV library, and Canny Edge Detection in the context of visual monitoring systems is discussed. This literature review establishes a strong foundation for subsequent chapters by offering valuable insights into the glass bottle manufacturing industry and the technological advancements essential for effective visual monitoring and inspection.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter will be presented the methodology adopted for the design and development of the visual monitoring and inspection system using VB.NET interface and CNNs based defect detection model for the glass bottle manufacturing industry. The methodology covers the system architecture, hardware components, and software implementation, offering a comprehensive insight into the approach employed to develop this innovative solution for automated defect monitoring in the glass bottle manufacturing industry.

The main objectives of this chapter are to delineate the systematic steps taken to design and construct the visual monitoring system for the glass bottle manufacturing industry. By elucidating the chosen methodology, readers will acquire insights into the overall structure of the system, the selection of hardware components, and the software implementation process.

The methodology employed in this project aims to integrate a user-friendly Human Machine Interface (HMI) as a conveyor control module with the defect detection module utilizing a deep learning algorithm to create an efficient visual monitoring system. Accurate identification of proper non-defective bottles and defective bottles is achieved, efficiently controlling the production process through a user-friendly interface for operators.

To offer a comprehensive overview, this chapter is segmented into various sections. Initially, it introduces the system architecture and functionality, elucidating the overall structure and components of the visual monitoring system. Next, a detailed exploration of the hardware components is undertaken, with their individual roles and functionalities within the system being explained. The purpose of each component and how they contribute to controlling the conveyor system to replicate the production line environment. The usage of the Arduino Uno as microcontroller to receive control commands from user-interface, suitable DC motor, LCD display, 12C, power supply modules, relays, and conveyor belt are discussed.

Additionally, the software implementation aspect is elucidated, detailing the programming of the Arduino microcontroller to establish communication and execute motor control actions. Of paramount importance is the creation of the image recognition and defect detection module, powered by the CNNs algorithm, is executed using software tools such as Python, PyCharm, and Microsoft Visual Code for Python language programming. Moreover, a user-friendly interface is designed using VB.NET in Microsoft Visual Studio for easy system control. Additionally, a database using Microsoft Access is implemented to efficiently manage the data generated by the monitoring system.

Furthermore, the methodology encompasses research design, data collection, implementation, testing and validation, ethical considerations, and project management. It outlines the framework tailored for the industry's challenges and explains the data collection methods, including camera footages. Testing and validation ensure compliance with requirements, while ethical considerations address data privacy and security. Project management involves timelines, milestones, and resource allocation.

In summary, this chapter provides a thorough guide to the methodology employed in developing this project. By explaining the system architecture, hardware components, and software implementation, readers gain insight into the technical aspects of this project. The methodology lays the groundwork for upcoming chapters, where results and discussions analyze system performance and draw conclusions for future work.

3.2 Incorporating Societal and Global Considerations in System Development

In the development of the visual monitoring system for the glass bottle manufacturing industry, it is essential to address societal and global issues alongside sustainable development principles. By incorporating these considerations into the system design and architecture, we aim to ensure that the system aligns with sustainability goals and contributes to broader societal needs. System design and architecture decisions have a significant impact on the overall societal and environmental implications of the visual monitoring system. By integrating sustainable development principles into the design process, we can optimize resource utilization, minimize environmental impact, and promote social benefits.

Moreover, it is important to emphasize the significance of user-centered design principles in the development of the visual monitoring system. By actively involving stakeholders in the design process and considering their perspectives, we can ensure that the system is tailored to address specific societal challenges and contribute to sustainable development goals. Furthermore, user-centered design principles are essential in addressing specific societal challenges.

Technological advancements, such as cloud computing and data analytics, play a crucial role in enabling sustainable system design. These technologies offer opportunities for energy optimization, real-time monitoring, and data-driven decision-making, which are instrumental in achieving sustainability objectives.

By integrating societal and global considerations alongside sustainable development principles, the visual monitoring system aims to provide a robust and sustainable solution for the glass bottle manufacturing industry. The insights and recommendations discussed in this context will guide the development process, ensuring that the system maximizes its positive impact on society and the environment while addressing the challenges associated with industrial automation.

3.3 System Architecture and Modules

In elucidating the system architecture and functionality, a comprehensive overview of the project's structural design ad operational aspects is presented. The system architecture encompasses the integration of key components, including the mini conveyor, VB.NET Human Machine Interface (HMI), and the image recognition module powered by Convolutional Neural Networks (CNNs). Each component's role and functionality within the larger system are detailed, providing a holistic understanding of their interactions.

The mini conveyor serves as the physical apparatus responsible for transporting glass bottles through the monitoring process. Its size, speed, and compatibility with the glass manufacturing environment are meticulously considered to ensure optimal performance. Concurrently, the VB.NET HMI interface is designed to facilitate user interaction with the monitoring system. It provides real-time information display, control over the conveyor, and a platform for users to analyze and respond to monitoring results.

The image recognition module, utilizing CNNs, plays a pivotal role in defect detection. This module processes images captured by cameras on the conveyor, distinguishing between proper and defective glass bottles. The CNN-based predictive model, with three convolutional layers, enhances the accuracy and efficiency of the defect detection process. Additionally, a real-time colour recognition feature extracts colour details from captured frames, further refining defect categorization.

The integration of these components ensures a cohesive and efficient system architecture. The mini conveyor, VB.NET HMI interface, and CNN-based image recognition collectively contribute to the seamless operation of the monitoring system. This section provides a detailed exploration of their interplay, elucidating the overall functionality that enables real-time, automated monitoring and efficient segregation of defective products in the glass manufacturing process.

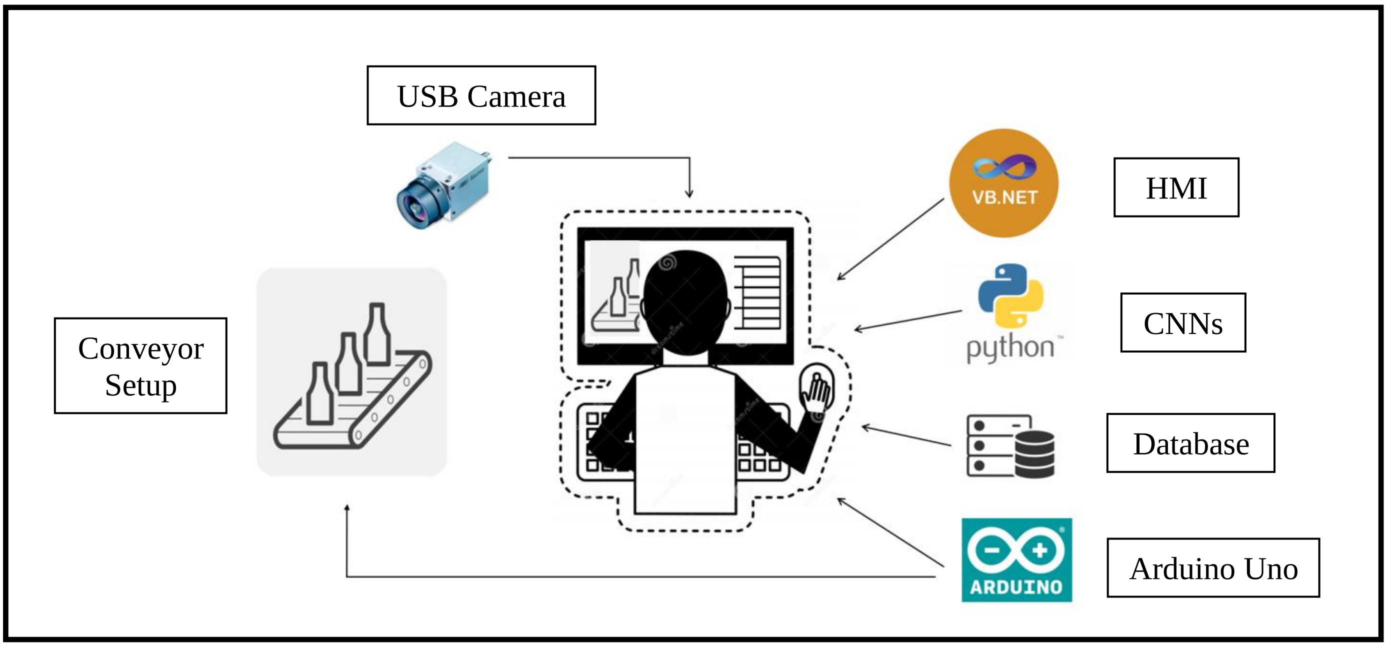


Figure 3-1 System Architecture and Component

3.3.1 Modules Description and Functionality

Within this section, the system architecture is intricately explored, elucidating the roles and operations of individual components. This section would outline what each component is responsible for, how it interacts with other components, and its specific functionalities in contributing to the overall operation of the system.

I. Image Acquisition Module

Responsible for capturing images of glass bottles on the conveyor belt through USB camera connected to a PC. Ensuring proper lighting, clear backgrounds, and full coverage of the glass bottle, it guarantees high-quality image acquisition. The acquired images are then seamlessly transferred to the subsequent modules for further processing.

II. Image Preprocessing Module

The Image Preprocessing Module is integral for refining captured images before Convolutional Neural Networks (CNNs) analysis. Key tasks, including resizing to 60x60 pixels for standardized processing, normalization for uniform data distribution, and grayscale enhancement for improved clarity, contribute to optimal data quality. This module ensures compatibility and enhances feature extraction for CNNs. Overall, it transforms raw captures into a standardized format, optimizing defect detection in glass bottles during subsequent CNN-based analysis on the manufacturing line.

III. Convolutional Neural Network (CNN) Module

The CNN Module serves as the heart of the system, leveraging deep learning techniques to scrutinize pre-processed images for effective bottle defect detection. Trained on labeled data, this module analyzes preprocessed images and identifies defects such as defect or proper. The CNN architecture consists of multiple convolutional and pooling layers, extracting intricate features and patterns for accurate defect classification. The trained model's high accuracy enhances the system's capability to distinguish between defective and proper glass bottles, contributing to improved quality control in the manufacturing process.

IV. User Interface Module

Developed using Visual Basic (VB.NET), this module offers a user-friendly interface for operators to interact with the system. It displays inspection results, real-time conveyor running animation and motor status indication, triggers control actions using serial communication with Arduino, to start and stop the conveyor. The intuitive interface enhances user engagement and facilitates seamless monitoring and management of the glass bottle manufacturing process.

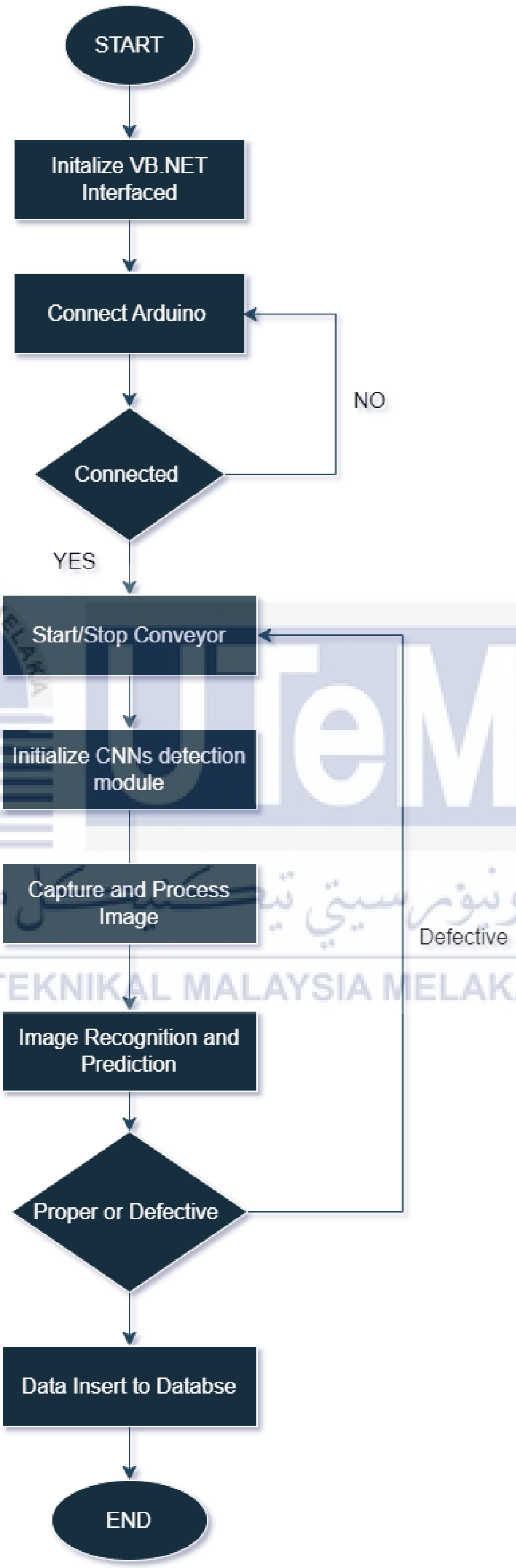
V. Microcontroller Module

The Microcontroller Module, powered by Arduino Uno, plays a crucial role in the system's control mechanism. Utilizing Serial Communication (SerialCom), it interfaces with the conveyor system, facilitating precise control and coordination. This module enables seamless communication between the central processing unit and the conveyor, ensuring efficient operations and synchronized actions.

VI. Database Module

This module manages the storage and retrieval of inspection data received from the CNN defect detection module. It is intricately connected to the VB.NET interface, enabling the display of inspection results for operators to monitor the operations. The database module ensures the proper organization and storage of inspection results, facilitating future analysis and serving as a valuable reference for system performance and historical data.

3.3.2 System Operation Flowchart



3.4 Project Development Process

The project development process begins with an Introduction, defining project objectives and the problem statement. This leads to a Literature Review and Research phase, exploring relevant articles on glass manufacturing, visual monitoring and inspection, and image recognition for defect detection. The subsequent Project Design and Methodology phase involves designing the hardware and software components such as Arduino circuit, mini conveyor, VB.NET HMI interface, and CNNs predictive model. The Testing and Troubleshooting phase ensure the reliability of each component. This phase aims to identify and address any potential issues or glitches that may arise during testing, ensuring the reliability and functionality of the integrated system. The Data Analysis and Experimentation phase involves testing various glass bottles for defect detection validation. Finally, the Conclusion phase summarizes findings, reflecting on achievements, challenges, and suggesting areas for future improvement. The flowchart in Figure 3.1 outlines a systematic development approach for the tailored visual monitoring and inspection system in the glass bottle manufacturing industry.

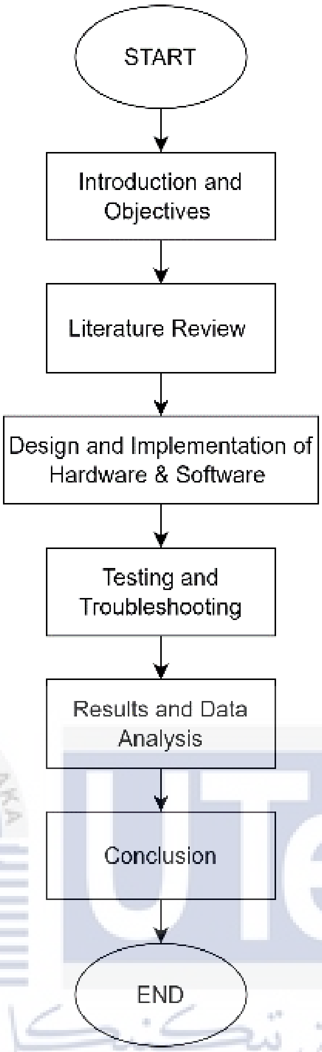


Figure 3-2 Project flowchart

3.5 Process of Conveyor Setup Development

In developing and integrating a mini conveyor for this project, a systematic approach is followed. Initially, the process incorporates 3D modelling techniques to visualize and refine the design before actual construction. This step ensures that the mini conveyor aligns seamlessly with the specified requirements, encompassing size, speed, and functionality. The selection of materials involves choosing robust and durable components, including a suitable DC motor, high-quality belts, bearings, and rollers. Upon establishing the design, the mini conveyor is constructed and assembled. Following the construction and development phases, the mini conveyor and VB.NET interface is integrated for proper communication and synchronization. Thorough testing and troubleshooting are then conducted to validate the integrated system's effectiveness.

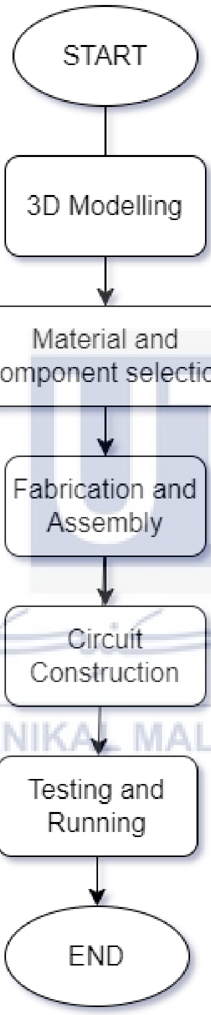


Figure 3-3 Conveyor Setup Flowchart

Figure 3-4 Mini Conveyor Setup

3.5.1 3D Modelling of Conveyor

The design of the conveyor has been carefully designed and dimensioned using Tinkercad. The conveyor structure shown in Figure 3-5 measures 50 cm in length, 18 cm in width, and 20 cm in height in total, (L x W x H). The design incorporates a complete structure that includes bearing, rollers, belt, and a DC Motor.

#

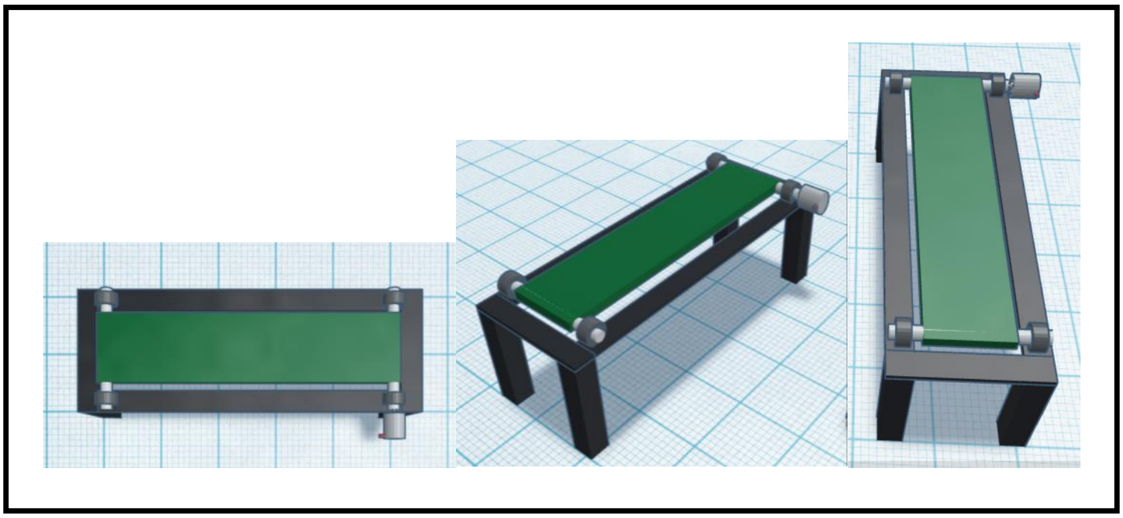


Figure 3-5 3D Design using ThinkerCad

3.5.2 Fabrication of the Conveyor

The fabrication of the conveyor system involves meticulous material selection, precise welding work, and careful assembly of components such as the conveyor belt, bearings, rollers, and DC motor. Attention to detail ensures durability and optimal functionality. The process includes secure attachment of bearings and rollers, precise integration of the DC motor, and rigorous testing to verify structural integrity.

Figure 3-6 Mini Conveyor Fabrication

3.5.3 Circuit Construction for Conveyor

This section outlines the construction of the circuit that governs the conveyor system using an Arduino Uno for serial communication. The setup involves essential components such as the 12V 10A Switch Mode Power Supply box (SMPS) , 12V 7A relay, a buck converter of 5V & 12V, 16x2 LCD display, 12C, 12V DC motor. The Motor Control Pin (motorPin - Pin 3) is crucial in the conveyor circuit as it is directly linked to a relay. This connection allows for the seamless activation and deactivation of the conveyor motor, ensuring precise control over its operational status. The Arduino Uno is programmed tointerprets commands ('A' to start and 'B' to stop) from the Visual Basic (VB) .NET interface, controlling the relay and the DC motor accordingly. Real-time status updates are displayed on the connected LCD, ensuring effective communication and operation throughout the bottle inspection process.

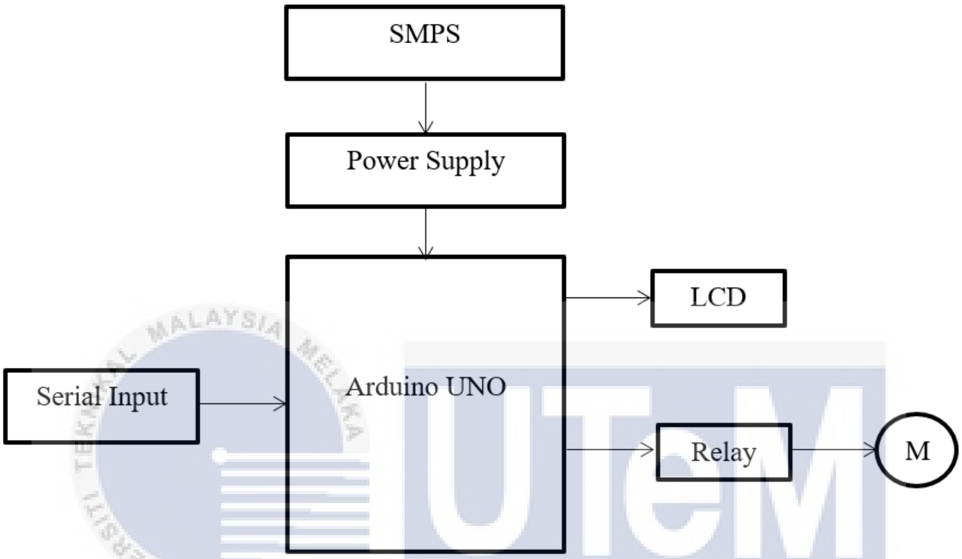


Figure 3-7 Circuit Block Diagram

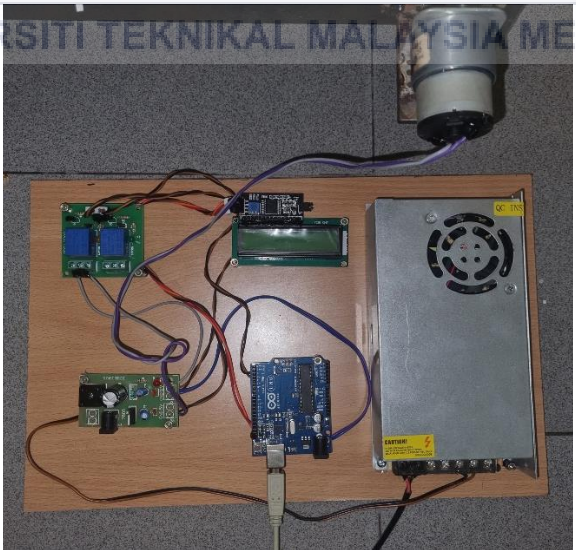


Figure 3-8 Circuit Construction

3.6 Process of VB.NET Interface Creation

In the development process of the VB.NET interface for this project, a systematic and methodical approach is employed. The initial step involves designing the user-friendly interface, considering the specific needs of the operators. The design includes features for displaying inspection results, controlling the conveyor, and triggering necessary actions. Once the design is established, the construction and coding phases commence. The interface is coded using Visual Basic (VB), ensuring a seamless and efficient interaction with the underlying system. Rigorous testing ensures reliability, and upon successful completion, the intuitive interface is ready for deployment, empowering operators to manage the glass bottle inspection system effectively.

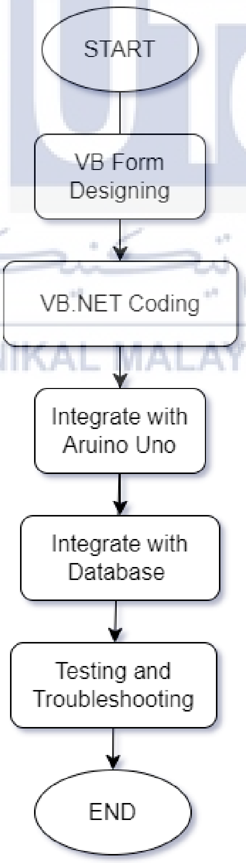


Figure 3-9 VB.NET Interface Process Flowchart

3.6.1 VB.NET Interface Design

The VB.NET interface is structured with two forms to facilitate user interaction seamlessly. The first form in Figure 3-10 incorporates Arduino serial communication controls, conveying the status of the connection, HMI controls for initiating and halting the conveyor via buttons, real-time animation depicting the conveyor's status, and motor status indicators. Additionally, a button directs to the second form for generating reports.

The second form (Report Form or Form2) in Figure 3 is linked to the database, housing the inspection results categorized as proper and defective bottles obtained from the CNN model for monitoring purposes. A back button allows users to return to Form 1, ensuring a cohesive and user-friendly interface.

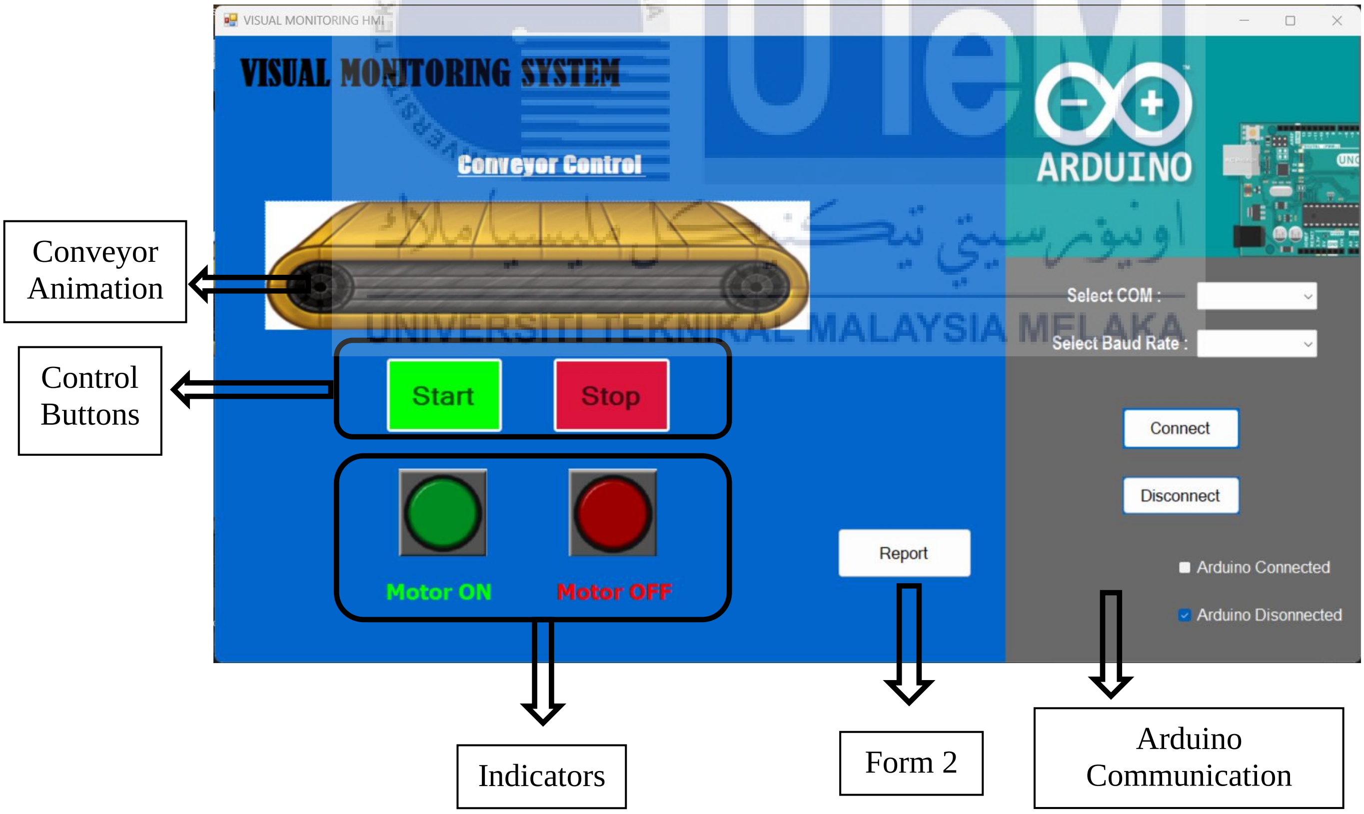


Figure 3-10 VB Form 1 : HMI

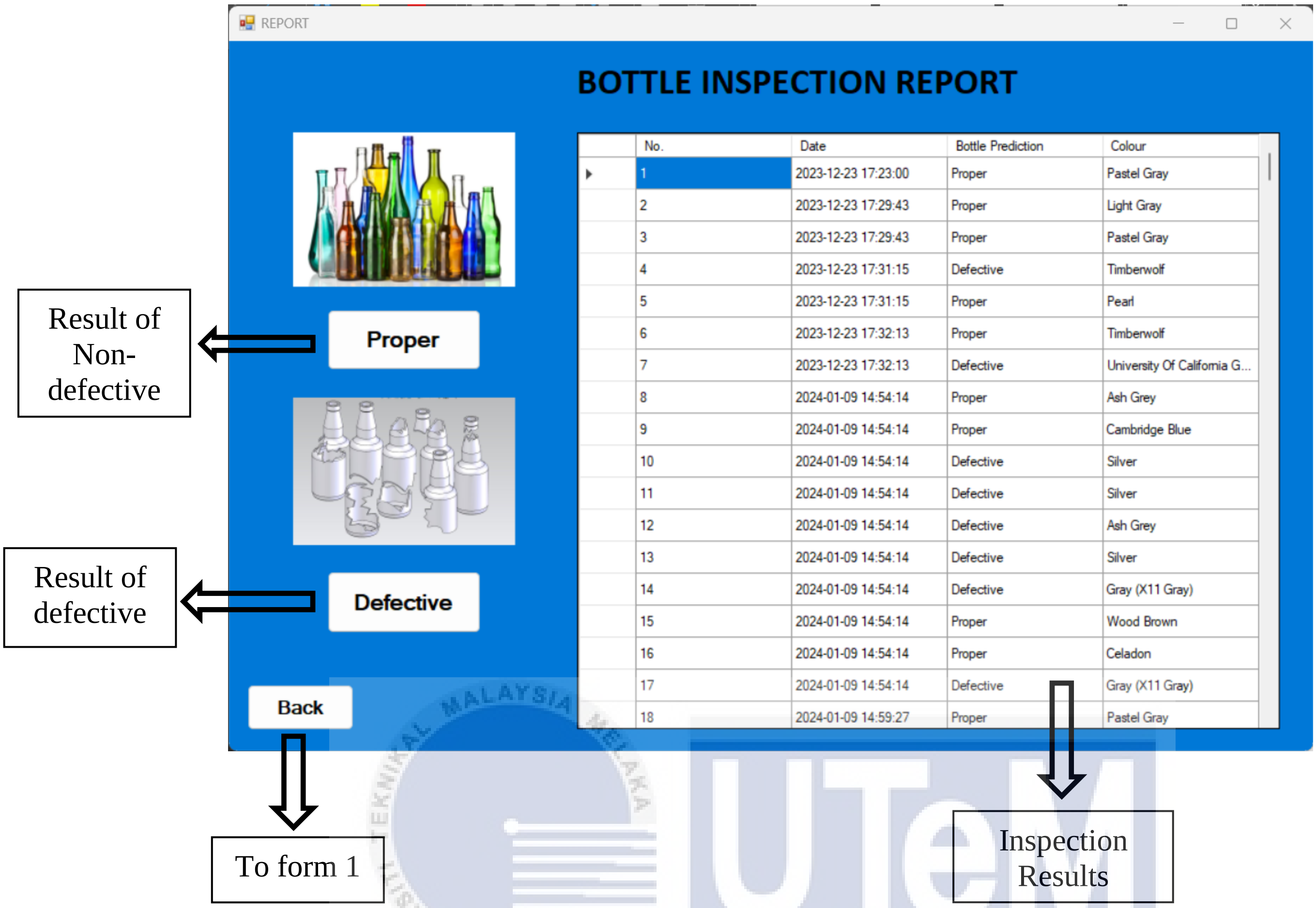


Figure 3-11 VB Form 2 : Report

3.7 Process of CNNs Deep Learning Defect Detection Module Development

This development process unfolds across several crucial modules. The first module emphasizes Image Preprocessing and CNN Dataset Preparation for Defective Product Classification. Following this, the second module focuses on constructing a CNN Predictive Model for Product Classification. The final stage involves the real-time application of CNN Classification in Glass Bottle Inspection, seamlessly integrating the trained model to detect defects during live video analysis. This structured three-module approach ensures a systematic and effective development process, covering data preparation, model creation, and real-time defect detection capabilities using CNNs.

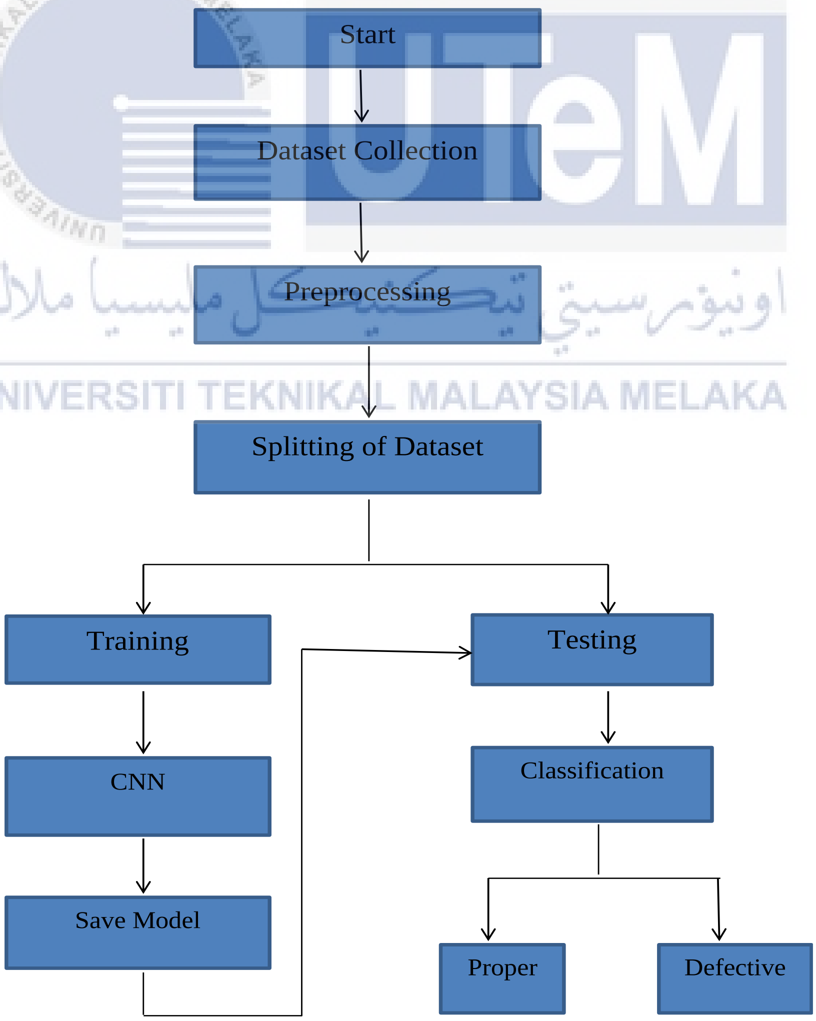


Figure 3-12 CNN Model Block diagram

3.7.1 Image Preprocessing and CNN Dataset Preparation for Defective Product Classification

The objective is to establish a robust dataset for training a CNN to identify proprer and defective products.

Key Steps:

1) **Image Loading:** Import images from categorized directories (Proper, Defective).
2) **Transformation:** Convert images to grayscale and resize them uniformly (60x60 pixels).
3) **Data Organization:** Structure data into pairs of image arrays and corresponding labels, forming a cohesive dataset.
4) **Shuffling:** Introduce randomness by shuffling the dataset to mitigate model biases.
5) **NumPy Conversion:** Transform shuffled data into NumPy arrays (features and labels).
6) **Data Preservation:** Save the prepared dataset using pickle for future accessibility.

3.7.2 Construction of CNN Predictive Model for Product Classification

Objective is to develop a CNN model capable of accurately classifying products as Proper or Defective based on the prepared dataset.

Key Phases:

1) **Load Prepared Dataset:** Retrieve features (X) and labels (y) using pickle.
2) **Normalization:** Normalize pixel values by dividing by 255.

3) **Reshape Data:** Adjust data shape for optimal CNN input.

4) **Model Architecture:** Construct a CNN model integrating convolutional layers, max-pooling layers, and dense layers.

5) **Compilation:** Compile the model with specified settings (optimizer, loss function, metrics).

6) **Training:** Train the model on the prepared dataset, specifying epochs and batch size.

7) **Evaluation:** Assess model accuracy on the testing set and report results.

8) **Model Preservation:** Save the trained model for subsequent use.

3.7.3 Real-time application of CNN Classification in Glass Bottle Inspection

Implement a real-time system for inspecting glass bottles, integrating the trained CNN model for accurate classification.

Key Stages:

1) **Library and Database Configuration:** Import essential libraries and establish a database connection for logging inspection outcomes.

2) **Colour Recognition Initialization:** Configure colour recognition based on an external CSV file.

3) **Image Processing and Model Loading:** Define functions for image processing and load the pre-trained CNN model.

4) **Webcam Setup:** Initialize a webcam to facilitate real-time video feed.

5) **Real-time Image Capture and Processing Loop:** Continuously capture frames, apply image processing techniques, and leverage the CNN model for classification.

6) **Result Logging:** Log relevant information, including date, classification, and colour recognition, into the connected database.
7) **Live Visualization:** Display processed frames in real-time for immediate feedback.
8) **User Interaction:** Allow the user to exit the system seamlessly.
9) **Resource Cleanup:** Release resources such as the webcam and close windows once the process is complete.

3.8 Hardware Component

This section offers an in-depth examination of the hardware components employed in the visual monitoring system. Each component assumes a distinct role in the control mechanism, contributing indispensably to the overall functionality of the system. The subsequent subsections provide detailed insights into the pivotal hardware elements integral to the system's operation.

3.8.1 Arduino Uno

The Arduino Uno microcontroller board consists of a ceramic resonator (CSTCE16M0V53-R0) with 16 MHz, a total of 22 pins, 6 PWM outputs (included in the digital pins, a USB port, also a power connector. It also contains a reset button, an internal LED attached to pin 13, and an ICSP connector for externally programming the microcontroller. To get started, it must link to a USB cable for computer connection, or a suitable battery or an AC-to-DC converter for power supply. An external DC power source (7-12V) or USB port can be used to power the Arduino Uno though it runs on a 5V DC voltage. The microcontroller board will handle the processing and control functions of the PMSM motors by taking input from sensors.

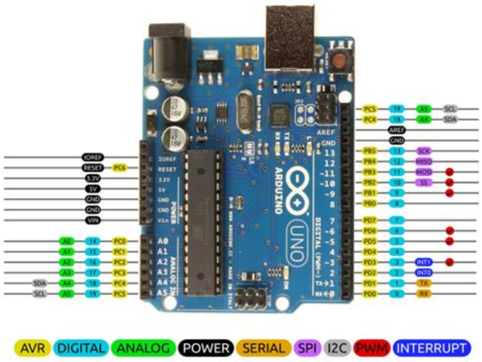


Figure 3-13 Arduino UNO pinout diagram

Table 3.1 Arduino UNO hardware specifications

Microcontroller:	ATmega328P
Operating Voltage:	5V
Input Voltage (recommended):	7-12V
Input Voltage (limit):	6-20V
Digital I/O Pins:	14 (0 – 13) (of which 6 provide PWM output)
PWM Digital I/O Pins:	6 (3, 5, 6, 9, 11)
Analog Input Pins:	6 (A0 – A5)
DC Current per I/O Pin:	20 mA
DC current for 3.3V Pin:	50 mA
Flash Memory:	32 KB of which 0.5 KB used for Bootloader)
SRAM:	2 KB (ATmega328P)
EEPROM:	1 KB
Clock Speed:	16 MHz
LED_BUILTIN:	13
Length and Width	68.6 mm × 53.4 mm
Weight	25g

3.8.2 JGB37-520 DC Motor

The JGB37-520 High Torque High-Speed Metal Gear DC Gear Motor showcased in figure 3.12 is a durable and dependable motor renowned for its robustness, ability to generate high torque, and utilization of metal gears. It operates efficiently within a rated voltage range of 12V-15V DC, drawing a rated current of 0.3A. With an extensive range of output speeds spanning from 12 to 960 RPM, this motor is well-suited for applications necessitating substantial power and speed. It possesses a rated output torque that varies from 0.17 to 13.6 kg.cm and a maximum torque of 31 kg.cm, enabling it to effectively handle heavy loads. The implementation of metal gears in the motor enhances its durability and strength, ensuring consistent performance even under demanding operating conditions. Additionally, the motor incorporates thermal protection mechanisms to prevent overheating during extended periods of use. Whether it is employed in robotics, electric vehicles, or industrial machinery, the JGB37-520 motor proves to be an outstanding choice due to its ability to deliver high power, speed, and reliability.

Figure 3-14 JGB37-520 DC Motor

3.8.3 12V 7A Relay

The 12V 7A Relay is a compact, versatile, and highly reliable electromechanical switching device designed for printed circuit board (PCB) mounting. It operates on 12 volts and provides a Single Pole, Double Throw (SPDT) configuration. This type of relay is often referred to as a "sugar cube" relay due to its small and square-shaped design. It is a fundamental component used to control the flow of electrical current in a wide range of applications, ensuring precision and efficiency in various industries.

The SPDT configuration of this relay means it has one common terminal, one normally open (NO) terminal, and one normally closed (NC) terminal. The relay works by using a coil to generate a magnetic field, which in turn mechanically toggles the state of the electrical contacts. When the coil is energized with a 12V power supply, it can switch the common terminal's connection between the NO and NC terminals, providing a versatile switching action.

Figure 3-15 12V 7A Relay

3.8.4 Switch Mode Power Supply (SMPS)

The 12V 10A SMPS (Switched Mode Power Supply) is a power supply unit that is both compact and technologically advanced. Its primary purpose is to convert electrical power from its input source to a stable 12-volt direct current (DC) output in an efficient manner. Through the utilization of cutting-edge switching technology, this switch-mode power supply (SMPS) guarantees a dependable and regulated power supply for a wide range of electronic equipment. Because of its narrow form factor and high current capacity, it is a versatile solution that may be utilized for applications that require a power source that is both reliable and effective.

Figure 3-16 12V 10A SMPS

3.8.5 16x2 LCD

This is an LCD Display designed for E-blocks. It is a 16 character, 2-line alphanumeric LCD display connected to a single 9-way D-type connector. This allows the device to be connected to most E-Block I/O ports. The LCD display requires data in a serial format, which is detailed in the user guide below. The display also requires a 5V power supply. Please take care not to exceed 5V, as this will cause damage to the device. The 5V is best generated from the E-blocks Multi programmer or a 5V fixed regulated power supply.

Figure 3-17 16x2 LCD

3.8.6 12C

The particular LCD I2C module that is being addressed here often consists of an I2C backpack in addition to a conventional HD44780-compatible television. Character-based and graphical text display are both possible with the HD44780 compatible LCD, and the I2C backpack has an I2C expander, which reduces the number of pins required for communication to just two (SDA and SCL). The connecting process is simplified as a result of this integration, and communication with microcontrollers that use the I2C protocol is made easier.

Figure 3-18 12C

3.8.7 Conveyor Belt

The PVC (Polyvinyl Chloride) conveyor belt is a sort of conveyor belt that is predominantly made out of PVC material. This form of conveyor belt is well-known for its longevity, flexibility, and resilience to a wide range of environmental conditions. It is common practice in industrial settings to make use of PVC conveyor belts for the purpose of transporting goods, materials, and products along conveyor systems in a manner that is both efficient and dependable. These belts are extremely important in a variety of industries, including packaging, logistics, and manufacturing, among others.

Figure 3-19 PCV Conveyor Belt

3.8.8 USB Camera

The Logitech C270 HD Webcam is a compact and dependable device known for its high-definition video and clear audio. It is suitable for various applications like video conferencing, live streaming, and content creation. With its built-in microphone, automatic light correction, and autofocus, it ensures excellent image quality. The webcam offers an affordable and convenient option for high-quality video communication and content production, including features like face tracking and motion detection. It was chosen for colour detection in the project. Figure 3.20 displays the webcam.

Figure 3-20 Logitect C270 HD Webcam

3.9 Software

This section describes the software components utilized in the development of the visual monitoring system for the glass bottle industry's industrial automation system. The software development aspect prioritizes the VB.NET programming language Python programming language as the primary coding and system development instrument. Python is widely used in the field of deep learning, and it's particularly popular for working with Convolutional Neural Networks (CNNs). VB.NET provides a comprehensive platform for constructing user-friendly interfaces, managing data processing, and facilitating communication between hardware components.

3.9.1 Microsoft Visual Studio

Microsoft Visual Studio is a versatile integrated development environment (IDE) widely used for software development. It supports various programming languages and facilitates designing, debugging, and deploying applications across platforms. With features like advanced code editing and seamless integration with Microsoft Azure, Visual Studio streamlines workflows for developers, enabling the creation of applications for desktop, web, mobile, and the cloud. Its continuous updates reflect Microsoft's commitment to providing a cutting-edge development environment, making Visual Studio essential for efficient software development.

Figure 3-21 Microsoft Visual Stusio

3.9.2 Arduino IDE

The software is specialized for programming Arduino microcontrollers like Uno, Raspberry Pi, and Nano, operating as an Integrated Development Environment (IDE). It's tailored for electric car traction system firmware in C/C++. The IDE facilitates seamless communication with Arduino hardware, offers debugging tools through a serial monitor, and simplifies the control mechanism for easy code writing, building, and uploading. Users can transfer and simulate code in Proteus circuit design, as depicted in Figure 3.22.

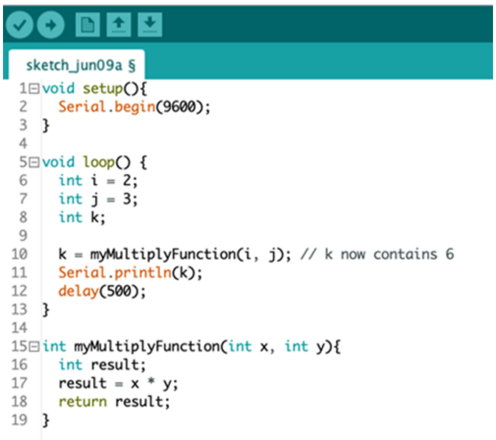


Figure 3-22 A sample of coding in Arduino IDE

3.9.3 Pyhton

Python, a versatile and widely-used programming language, serves as a cornerstone for numerous applications, including the implementation of Convolutional Neural Networks (CNNs) for defect detection. Python's popularity in the realm of deep learning stems from its simplicity, readability, and an extensive ecosystem of libraries and frameworks tailored for machine learning tasks.

In many fields, including web development, data analysis, artificial intelligence, and scientific computing, Python is a high-level, interpreted programming language that isutilised extensively. It was developed by Guido van Rossum and published in 1991; since then, it has grown in popularity as a result of its clarity, readability, and sizeable library. Python is renowned for its succinct and clear syntax, which makes it simple to learn and comprehend. It improves code readability by defining code chunks with indentation. Programming paradigms including procedural, object-oriented, and functional programming are all supported by the language.

Python has a robust ecosystem of libraries and frameworks, which is one of its main advantages. Thousands of open-source libraries offering solutions for a wide range of tasks are hosted by the Python Package Index (PyPI). Popular libraries include TensorFlow and PyTorch for machine learning and deep learning, Pandas for data analysis and manipulation, Matplotlib for data visualisation, and NumPy for numerical computing.

Python's cross-platform interoperability allows programmers to create code that can be executed on a variety of operating systems, including Windows, macOS, and Linux. Developers can interface with various languages and technologies because to its robust integration capabilities. Python, for instance, can be used for scripting jobs within bigger software systems or integrated in C/C++ programmes.

A thriving and active community supports the Python software development ecosystem. Through online forums, discussion groups, and tutorials, this community supports language users, builds and maintains libraries, and contributes to the language's growth. Additionally, Python provides integrated development environments (IDEs) and development tools that facilitate the writing, debugging, and testing of code. PyCharm, Visual Studio Code, and Jupyter Notebook are examples of well-known IDEs.

Figure 3-23 Pyhton Software

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This section presents the outcomes of the implemented methodologies, including project outcome, stimulation, resutls, and analysis. The results align with the initial objectives and work scope of the project. The aim is to demonstrate the achievement of project objectives through the presented results and analyses. The following sections provide a detailed analysis and interpretation of the outcomes, evaluating their effectiveness, efficiency, and relevance to the project. By showcasing these results, it becomes evident how the implemented methodologies have successfully contributed to the development of a visual monitoring system for the glass bottle industry's industrial automation system.

In the results section, the key findings and outcomes of glass bottle shape defect detection system are presented. This includes information such as the accuracy of the prediction, the efficiency of the system in any relevant metrics or statistics that highlight the system's performance.

The results obtained encompass an evaluation of the system's effectiveness in classifying defective and non-defective glass bottles of varied types and shapes. A key focus lies in preventing false predictions by the system. These findings contribute to a comprehensive assessment, providing both quantitative and qualitative insights into the system's capabilities and its impact on the inspection and monitoring process.

Moreover, the examination extends beyond shape defects, encompassing an in-depth analysis of colour matching recognition. The results also encapsulate data derived from testing the system's response to various colour scenarios, further enhancing the system's overall robustness and versatility in defect detection.

In the analysis and discussion, the results will be meticulously interpreted and analyzed to provide in-depth insights. This involves delving into the implications and significance of the findings within the specific context of the glass bottle defect detection system. The discussion will explore various aspects, encompassing the strengths and limitations of the system, potential avenues for enhancement, and comparisons with existing defect detection systems or alternative technologies in the manufacturing industry.

Overall the results and discussions section offers a comprehensive analysis of the glass bottle defect detection system, validating its effectiveness, revealing strengths and weaknesses, and providing valuable insights for potential enhancements in manufacturing applications.

4.2 Project Design Outcome

The project design attains its goals with a successful integration of both software and hardware components for the development of a visual monitoring system tailored to the glass bottle manufacturing industry. The VB.NET interface empowers operators to control conveyor operations and seamlessly access inspection reports, enhancing operational control and result visibility. Figure 4.1 below shows the developed conveyor system with a inspection camera and Figure 4.2 below shows VB.NET interface with control action and inspection results.

Figure 4-1 Design Outcome

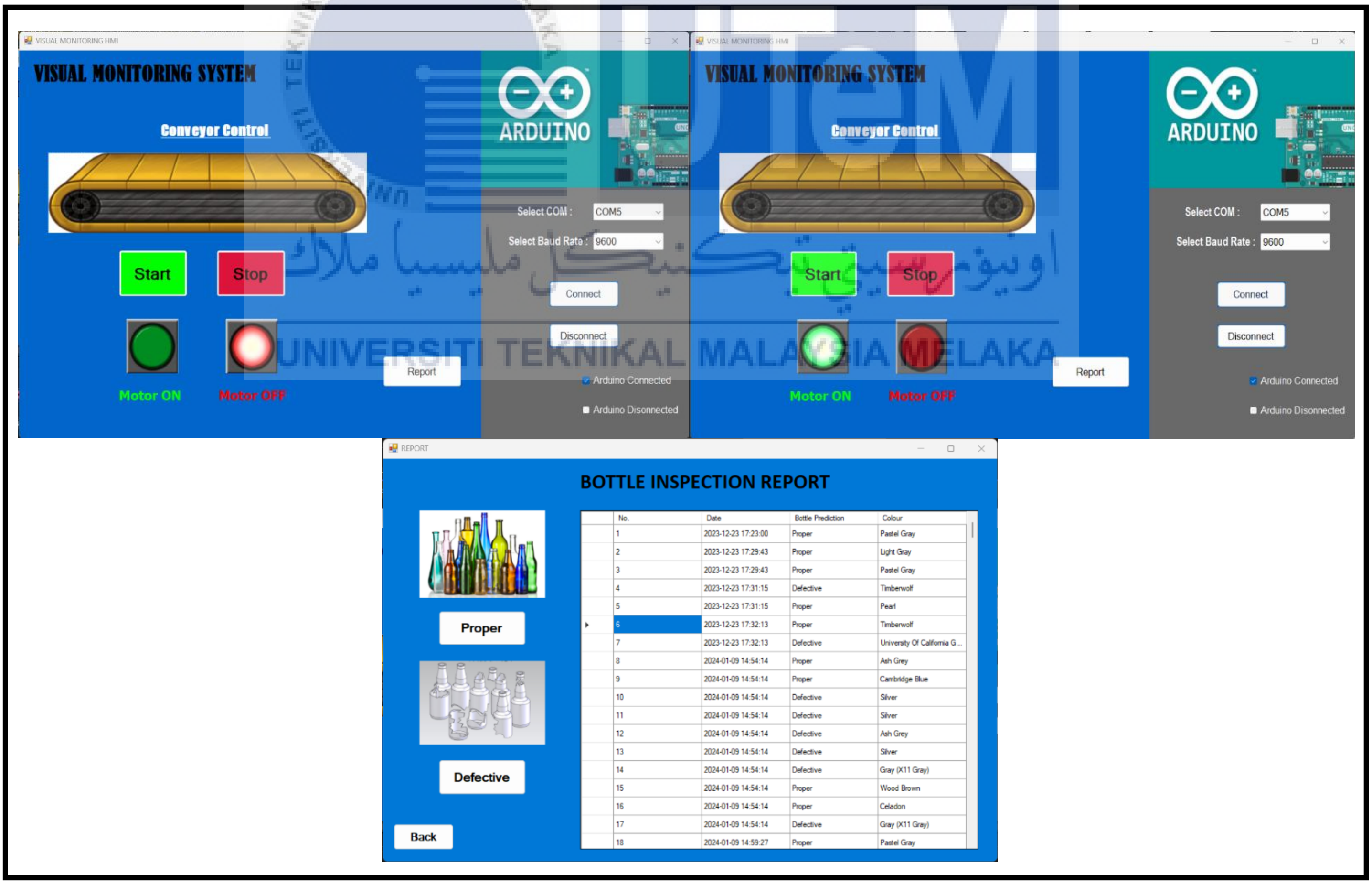


Figure 4-2 VB.NET Interface Outcome

4.3 Results and Data Collection

In preparation for this section, the system underwent training using a dataset comprising four types of glass bottles distinguished by shape and colour. Within each bottle type, the defective dataset was curated to include 2 or 3 distinct shape defects localized in the mouth and neck areas. In parallel, the proper dataset was trained with non-defective, proper glass bottles, ensuring representative coverage for each bottle type. This meticulous training approach establishes a foundation for robust defect detection and classification, and the subsequent results and data collection phase aims to elucidate the system's performance across these varied scenarios.

Following the training process, the subsequent phase involved systematic results and data collection for each each bottle type, considering multiple shape defects, particularly in the neck and mouth areas. To validate the system's effectiveness and accuracy, additional data points were collected using untrained defective bottles for each specified bottle type. This inclusive and thorough data collection strategy provides a nuanced understanding of the system's capabilities, offering insights into its adaptability and limitation under diverse real-world conditions and defect scenarios.

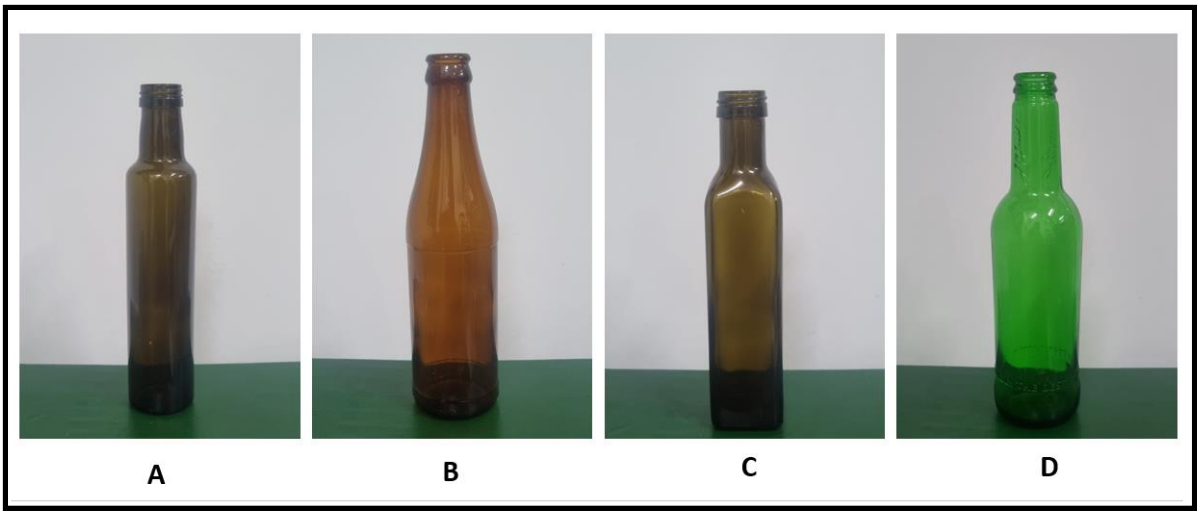


Figure 4-3 4 Type of Glass Bottles

Figure 4-4 Defective Glass Bottles

4.3.1 Glass Bottle Defect Detection Result

In this subsection, the outcomes of the glass bottle defect detection system are unveiled. The CNN predictive model showcases its adeptness in discerning between proper and defective bottles. A comprehensive assessment across various bottle types, encompassing a spectrum of shapes and colours, elucidates the system's performance. The successful identification of shape defects in the neck and mouth areas underscores the system's efficacy in classifying defects within the meticulously trained dataset. Furthermore, results obtained from untrained defective bottles illuminate the system's adaptability and robustness, showcasing its effectiveness in detecting defects across a myriad of scenarios.

The table 4.1 below showcases the results obtained from the database analysis.

Table 4.1 Glass Bottle Prediction Result

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1206	2023-1-9 17:23:00	Proper	Lime Green	
1207	2023-1-9 17:23:00	Defective	Lime Green	
1208	2023-1-9 17:29:43	Defective	Brown (traditional)	
1209	2023-1-9 17:29:43	Proper	Ginger	
1210	2023-1-9 17:31:15	Defective	Timberwolf	
1211	2023-1-9 17:31:47	Defective	Mahogany	

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1212	2023-1-9 17:32:13	Proper	Timberwolf	
1213	2023-1-9 17:32:33	Defective	Mahogany	
1214	2024-01-09 17:54:14	Defective	Neon Green	
1215	2024-01-09 17:54:30	Defective	Braun (traditional)	
1216	2024-01-09 17:54:36	Defective	Black	
1217	2024-01-09 17:56:02	Defective	Chocolate (Traditional)	
1218	2024-01-09 17:57:00	Defective	Lime Green	

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1219	2024-01-09 17:57:14	Defective	Caput Martuur	
1220	2024-01-09 17:57:40	Proper	Caput Mortuur	
1221	2024-01-09 18:13:14	Defective	Wood Brown	
1222	2024-01-09 18:16:55	Proper	Wood braun	
1223	2024-01-09 18:24:14	Defective	Chocolate (tradisional)	

4.3.2 Glass Bottle Defect Detection Result In Different Settings

To achieve a more comprehensive and critical understanding, additional results were collected using the same glass bottles but under varied conditions, including different lighting scenarios, camera angles, and backdrops. This meticulous data collection process aims to not only reaffirm the effectiveness of the system's performance but also to uncover potential limitations and weaknesses within the proposed system. By subjecting the system to diverse environmental factors, the evaluation seeks to provide valuable insights into the system's adaptability and areas for improvement, contributing to a more robust and reliable glass bottle defect detection solution. The table 4.2 below howcases the result obtained from database with different settings and also the false prediction highligted in red.

Table 4.2 Bottle Prediction Result with Different Settings

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1206	2023-1-10 13:20:00	Proper	Caribbean Green	
1207	2023-1-10 13:23:00	Proper	Lime Green	
1208	2023-1-10 13:23:13	Defective	Café Noir	

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1209	2023-1-10 13:25:31	Defective	Ginger	
1210	2023-1-10 13:27:12	Defective	Timberwolf	
1211	2023-1-10 13:31:47	Defective	Mahogany	
1212	2023-1-10 13:32:03	Defective	Timberwolf	
1213	2023-1-10 13:42:33	Defective	Mahogany	
1214	2024-01-09 13:44:11	Defective	Malachite	

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1215	2024-01-10 13:50:00	Defective	Braun (traditional)	
1216	2024-01-10 13:52:33	Defective	Black bean	
1217	2024-01-09 14:00:02	Defective	Chocolate (Traditional)	
1218	2024-01-10 14:16:00	Defective	Lime Green	
1219	2024-01-10 14:19:14	Defective	Black bean	
1220	2024-01-10 14:20:40	Defective	Caput Mortuur	
1221	2024-01-10 14:21:14	Defective	Black bean	

ID	date_t	bottle_result	bottle_colour	Glass Bottles
1222	2024-01-10 14:38:51	Proper	Blak bean	
1223	2024-01-10 14:40:23	Defective	Chocolate (traditional)	

4.4 Analysis and Discussion

In this pivotal section, an in-depth analysis and subsequent discussion unfold, meticulously scrutinizing the performance of the glass bottle defect detection system. The comprehensive evaluation navigates through two pivotal focal points, shedding light on various aspects of the system's functionality and outcomes. Furthermore, the discussion delves into the parameters that may cause false predictions, meticulously examining the factors taken into consideration. This examination aims to provide a nuanced understanding of the system's strengths and potential areas for improvement. Additionally, the section will thoroughly uncover the consistency of colour recognition features, elucidating how well the system performs in distinguishing variations in colour, a crucial aspect in defect detection.

4.4.1 Analysis of Bottle Prediction Results Under Similar Settings to Trained Conditions

This segment of the analysis intricately examines how the system performs when predicting bottle outcomes in scenarios closely resembling the conditions present during the training phase. The evaluation encompasses crucial metrics such as accuracy, precision, and recall, providing insights into the model's ability to generalize effectively and make accurate predictions within a familiar operational context.

Based on the results of the CNNs predictive model predictions for proper and defective glass bottles under settings similar to the trained conditions, the outcomes were highly accurate, with nearly all predictions aligning with the expected results. However, it's noteworthy that one false prediction was identified. This instance will be further investigated to discern the contributing factors and explore potential refinements to enhance the model's robustness.

The system's adeptness in maintaining accuracy under conditions akin to its training environment reflects its reliability. The identification of a single false prediction serves as a crucial learning point for system refinement, emphasizing the continuous improvement process in ensuring optimal defect detection performance. The ensuing discussion will delve into the specifics of this false prediction, aiming to glean valuable insights for system enhancement.

In the given context, the suitability of the CNN algorithm for scenarios closely resembling the trained conditions lies in its inherent strengths in capturing spatial hierarchies and patterns within images. CNNs (Convolutional Neural Networks) are particularly well-suited for image-related tasks due to their ability to automatically learn hierarchical representations of features.

Advantages of CNNs in Similar Settings to Trained Conditions:

Local Feature Learning: CNNs utilize convolutional layers for effective detection of local features across diverse image regions. This proves valuable in scenarios with consistent object structures, like glass bottles, observed in settings akin to training conditions.

Translation Invariance: CNNs, inherently translation-invariant, recognize patterns regardless of their image position. This is advantageous in processing images captured under similar settings, maintaining constant relative object positions.

Parameter Sharing: CNNs employ parameter sharing, using the same weights for different image regions. This proves beneficial in consistently recognizing similar patterns, aligning with requirements in settings resembling training conditions.

To conclude, based on the analysis the developed CNN algorithm proves to be highly valuable and reliable for detecting shape defects in the glass bottle manufacturing industry. This is because the conditions in production lines, such as lighting and camera positions, usually stay consistent during inspection process. The CNN's ability to learn local features and adapt to different positions makes it well-suited for these environments. Additionally, the rare occurrence of inaccuracies caused by foreign objects further ensures the model's accuracy, minimizing false predictions. The CNN algorithm's stability makes it an excellent choice for ensuring product quality and integrity in continuous production processes.

4.4.2 Analysis of Bottle Prediction Results Under Different Settings to Trained Conditions

The subsequent analysis delves into the system's performance under conditions that deviate from the initial training settings. This exploration aims to uncover variations in performance metrics, providing a comprehensive understanding of how the model's predictions unfold within an environment that aligns with its initial training conditions but introduces subtle differences.

The prediction results revealed a few instances of false predictions, primarily occurring on non-defective glass bottles. This can be attributed to varying lighting conditions, where reflections on the glass bottle surface may interfere with Canny Edge detection. Other influencing factors include camera angles, positions, conveyor speed, and image quality, all of which can significantly contribute to false predictions.

The presence of diverse lighting conditions poses a challenge, as variations in brightness and shadows can affect the model's ability to accurately detect edges using the Canny Edge detection technique. The reflective nature of glass surfaces can lead to inconsistencies in edge identification, contributing to occasional misclassifications.

In addition, factors such as camera angles and positions play a pivotal role. Changes in these parameters can alter the perspective from which the images are captured, impacting the model's ability to generalize effectively. The speed of the conveyor system also introduces variability, affecting the temporal aspect of image acquisition and potentially leading to misalignments in feature recognition.

Furthermore, the overall quality of the captured images is critical. Blurriness, noise, distortion, or inadequate resolution can hinder the model's capability to discern intricate details, contributing to errors in the classification process.

Addressing these challenges requires a multi-faceted approach, involving fine-tuning the model to adapt to different lighting conditions, optimizing edge detection techniques for reflective surfaces, and incorporating robustness against variations in camera angles, conveyor speed, and image quality. Continuous refinement of the model's parameters and training on diverse datasets can enhance its resilience, minimizing the impact of these factors on prediction accuracy.

4.5 Summary

In the culmination of this chapter, the implemented methodologies have been unfolded, encompassing project outcomes, simulations, results, and analyses. Aligned with the initial project objectives, the presented results illustrate the successful development of a visual monitoring system tailored for the glass bottle industry's industrial automation system.

The results showcased the system's performance in shape defect detection and colour matching recognition. The quantitative metrics underscored its efficiency and reliability, contributing to a comprehensive evaluation. The system's preventive measures against false predictions added an extra layer of reliability to its operational framework. Moving into the analysis and discussion section, the nuances of these results were unraveled, shedding light on the system's strengths and identifying areas for refinement. In summary, this chapter serves as a testament to the successful development of a visual monitoring system tailored for the glass bottle industry's industrial automation. In transition to the next chapter, the rich insights gained from this exploration will inform the recommendations and future directions, ensuring the continual evolution and optimization of our glass bottle defect detection system.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This thesis centers on the development of a visual monitoring system seamlessly integrated into an industrial automation framework, specifically tailored for the glass bottle industry. The outcomes of this research exhibit significant promise. Utilizing VB.NET and incorporating a deep learning algorithm for shape defect detection through Convolutional Neural Networks (CNNs), the system showcases its potential to revolutionize quality control and automation processes effectively.

Initial findings affirm the systems adeptness in accurately detecting and analyzing shape defects in real-time, offering invaluable insights for robust quality assurance. Additionally, the integration of a conveyor system enhances the efficiency of the production line by facilitating the smooth transportation of glass bottles throughout the manufacturing process.

The integration of the conveyor system and visual monitoring system delivers a myriad of benefits to manufacturers. It enables enhanced automation, streamlined workflows, and elevated product quality. The conveyor system ensures a seamless flow of materials, while the visual monitoring systems shape defect detection capabilities, powered by CNNs, allow for the identification of defective glass bottles. This integration empowers manufacturers to uphold stringent quality standards and consistent product specifications.

In conclusion, the synergy between the visual monitoring system and the conveyor system presents a holistic solution for the glass bottle industry. It facilitates efficient object handling, precise shape defect detection through CNNs, and reliable quality control. As technology progresses, continuous refinements and optimizations can be implemented to unlock the system's full potential, ultimately leading to heightened productivity and heightened customer satisfaction in the manufacturing process.

5.2 Potential for Commercialization

The integration of the visual monitoring system with a conveyor system not only signifies a breakthrough in the glass bottle industry but also unveils a substantial potential for commercialization. This groundbreaking solution, primarily driven by shape defect detection using Convolutional Neural Networks (CNNs), introduces numerous advantages, including refined quality control, optimized operational efficiency, and a distinct competitive edge. Offering a customizable and scalable framework adaptable to diverse manufacturing processes and facility scales, the system ensures cost-effectiveness. Embracing this technological advancement empowers manufacturers to elevate product quality, boost productivity, and curtail operational expenses.

Moreover, the proposed solution extends its applicability to the waste management industry, specifically in the classification of glass and plastic bottle waste. This concept can be harnessed for distinguishing between proper and defective bottles, facilitating efficient waste sorting for further reuse and recycling initiatives. This dual-industry application enhances the versatility and societal impact of the developed solution.

5.3 Future Works

To further enhance the accuracy and versatility of the shape defect detection system based on Convolutional Neural Networks (CNNs), and to advance the Human-Machine Interface (HMI), the following improvements can be considered:

I. **Advanced CNN Architectures:** Continuous exploration of advanced CNN architectures can significantly elevate the system's precision and speed in identifying and classifying shape defects. Refining the neural network structures can accommodate intricate patterns and variations, ensuring robust performance across diverse glass bottle shapes and defect types.

II. **Augmented Dataset for Training:** The system's efficacy can be further amplified by expanding and diversifying the dataset used for training. Incorporating a broader range of shape defects and variations ensures that the CNN is well-equipped to handle real-world scenarios with unparalleled accuracy.

III. **Real-time Adaptive Learning:** Integrating real-time adaptive learning mechanisms enables the system to dynamically adjust to evolving patterns and defects. This self-learning capability ensures continuous improvement and adaptability, particularly in dynamic manufacturing environments where new defect types may emerge.

IV. **Integration of Multi-Sensor Data:** In addition to CNN-based shape defect detection, integrating data from other sensors, such as depth sensors or infrared sensors, can provide complementary insights. This multi-sensor approach enhances the system's overall capabilities for comprehensive quality control and defect detection.

V. **Scalability and Modular Design:** Designing the system with scalability and modularity in mind ensures seamless integration into various manufacturing setups. A modular hardware design, compatibility with diverse conveyor systems, and flexibility in accommodating different bottle shapes and conveyor speeds contribute to the system's adaptability and widespread applicability.

VI. **Advanced HMI Interface (SCADA):** Implementing a more advanced Human-Machine Interface (HMI), such as a Supervisory Control and Data Acquisition (SCADA) system, can provide enhanced visualization, control, and monitoring capabilities. This ensures a more intuitive and user-friendly interface for operators, facilitating efficient management of the shape defect detection process.

By prioritizing these future enhancements, the visual monitoring system for glass bottle shape defect detection, coupled with an advanced HMI interface, can undergo continuous refinement, delivering enhanced value to manufacturers in the glass manufacturing industry.

REFERENCES

[1] M. Aadil Arshad, M. Kashif, N. Suresh Kale, and D. M. Ugle, “A Paper Review on ‘Glass Processing,’” 2017. [Online]. Available: www.ijsrd.com

[2] M. Kovačec, A. Pilipović, and N. Štefanić, “Improving the quality of glass containers production with plunger process control,” *CIRP J Manuf Sci Technol*, vol. 3, no. 4, pp. 304–310, 2010, doi: 10.1016/j.cirpj.2011.02.003.

[3] S. Padhee and D. Nandan, “Design of automated visual inspection system for beverage industry production line,” *Traitement du Signal*, vol. 38, no. 2, pp. 461–466, Apr. 2021, doi: 10.18280/ts.380225.

[4] S. Seker, “A new model for risk assessment in glass manufacturing using Risk Matrix based IVIF-TOPSIS method,” *Journal of Intelligent and Fuzzy Systems*, vol. 42, no. 1, 2022, doi: 10.3233/JIFS-2191210.

[5] N. Dhinakaran and M. Balachandar, “Hazard Identification and Risk Assessment in Glass Coater Plant,” *Environmental Science and Engineering*, vol. 1, no. 1, pp. 01–04, Mar. 2022, doi: 10.46632/ese/1/1/1.

[6] C. D. Westbroek, J. Bitting, M. Craglia, J. M. C. Azevedo, and J. M. Cullen, “Global material flow analysis of glass: From raw materials to end of life,” *J Ind Ecol*, vol. 25, no. 2, pp. 333–343, 2021, doi: https://doi.org/10.1111/jiec.13112.

[7] J. Bazroy, G. Roy, A. Sahai, and M. B. Soudarssanane, “Magnitude and Risk Factors of Injuries in a Glass Bottle Manufacturing Plant,” 2003.

[8] F. R. d’AMBROSIO ALFANO, B. I. Palella, G. Riccio, M. Bartalini, F. Strambi, and J. Malchaire, “Heat stress assessment in artistic glass units,” *Ind. Health*, vol. 56, no. 2, pp. 171–184, 2018.

[9] P. Oborski, “Integrated monitoring system of production processes,” *Management and Production Engineering Review*, vol. 7, no. 4, pp. 86–96, Dec. 2016, doi: 10.1515/mper-2016-0039.

[10] J. E. See, C. G. Drury, A. Speed, A. Williams, and N. Khalandi, “The role of visual inspection in the 21st century,” *The Role of Visual Inspection in the 21st Century. (Conference) | OSTI. GOV*. Jun. 2017.

[11] Pratik Maurya, Chinmay Gaikawad, and Suyash Salvi, “Visual Inspection for Industries,” *International Journal of Advanced Research in Science, Communication and Technology*, 2022, doi: 10.48175/ijarsct-2827.

[12] L. C. Dülger, E. Gültekin, H. İ. Çelik, L. C. Dülger, H. İ. Sünbül, and H. Kani, “IMAGE PROCESSING APPLICATIONS ON YARN CHARACTERISTICS AND FAULT INSPECTION.” [Online]. Available: https://www.researchgate.net/publication/333044032

[13] A. Shaikat, M. Rahman, and M. Khan, “An Image Processing Based Glass bottle Defect Detection System,” Jun. 2020.

[14] K. S. Kumar *et al.*, “Monitoring of diesel level in glass manufacturing industries with an automated fire extinguishing system,” *I-manag. J. Embed. Syst.*, vol. 4, no. 3, pp. 7–14, Oct. 2015.

[15] Y. Zhao, X. An, and N. Sun, “Virtual simulation experiment of the design and manufacture of a beer bottle-defect detection system.,” *Virtual Reality and Intelligent Hardware*, vol. 2, no. 4, pp. 354–367, Aug. 2020, doi: 10.1016/j.vrih.2020.07.002.

[16] C. Wang, “The application of machine vision in intelligent manufacturing,” *Highlights in Science, Engineering and Technology*, vol. 9, pp. 47–50, Sep. 2022.

[17] S. S, “INDUSTRIAL APPLICATION OF MACHINE VISION,” *Int J Res Eng Technol*, vol. Volume: 03, Jun. 2014, doi: 10.15623/ijret.2014.0319120.

[18] S. L. Diab, A. H. Bassiuny, M. Hussein, and R. Ahmed, “Designing a Visual Inspection System for Quality Characteristics Dimensions,” *Mechanical Engineering Research*, vol. 1, no. 1, Dec. 2011, doi: 10.5539/mer.v1n1p12.

[19] E. Hernandez-Molina, B. Ojeda-Magana, J. G. Robledo-Hernandez, and R. Ruelas, “Vision system prototype for inspection and monitoring with a smart camera,” *IEEE Latin America Transactions*, vol. 18, no. 9, 2020, doi: 10.1109/TLA.2020.9381804.

[20] R. Usamentiaga, M. A. Fernandez, A. F. Villan, and J. L. Carus, “Temperature monitoring for electrical substations using infrared thermography: Architecture for industrial internet of things,” *IEEE Trans. Industr. Inform.*, vol. 14, no. 12, pp. 5667–5677, Dec. 2018.

[21] R. Hamzeh, L. Thomas, J. Polzer, X. W. Xu, and H. Heinzel, “A sensor based monitoring system for real-time quality control: Semi-automatic arc welding case study,” in *Procedia Manufacturing*, Elsevier B.V., 2020, pp. 201–206. doi: 10.1016/j.promfg.2020.10.029.

[22] Y. Bayram, A. Ruege, P. Hagan, E. Sperry, and D. Cetnar, “Advanced Furnace Inspection and Monitoring Based on Radar Sensors,” 2016, pp. 117–132. doi: 10.1002/9781119282471.ch10.

[23] A. Shaikat, M. Rahman, S. Akter, and M. Khan, “Application of PLC and SCADA Based Real-Time Online Counter for Glass Bottle Manufacturing Industry in Bangladesh,” Jun. 2020.

[24] T. H. Tsai and C. S. Cheng, “Automatic Optical Inspection System for Wire Color Sequence Detection,” *Sensors*, vol. 22, no. 15, Aug. 2022, doi: 10.3390/s22155885.

[25] G. D’Emilia and D. Di Gasbarro, “Review of techniques for 2D camera calibration suitable for industrial vision systems,” *J. Phys. Conf. Ser.*, vol. 841, p. 12030, May 2017.

[26] T. Chen, Y. Wang, C. Xiao, and Q. M. J. Wu, “A machine vision apparatus and method for can-end inspection,” *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2055–2066, Sep. 2016.

[27] X. Zhou *et al.*, “Automated Visual Inspection of Glass Bottle Bottom With Saliency Detection and Template Matching,” *IEEE Trans Instrum Meas*, vol. 68, no. 11, pp. 4253–4267, Nov. 2019, doi: 10.1109/tim.2018.2886977.

[28] J. Wang, P. Fu, and R. X. Gao, “Machine vision intelligence for product defect inspection based on deep learning and Hough transform,” *J Manuf Syst*, vol. 51, pp. 52–60, Apr. 2019, doi: 10.1016/j.jmsy.2019.03.002.

[29] T. Benbarrad, M. Salhaoui, S. B. Kenitar, and M. Arioua, “Intelligent Machine Vision Model for Defective Product Inspection Based on Machine Learning,” *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, p. 7, Jan. 2021, doi: 10.3390/jsan10010007.

[30] B. Huang *et al.*, “Research and implementation of machine vision technologies for empty bottle inspection systems,” *Engineering Science and Technology, an International Journal*, vol. 21, no. 1, pp. 159–169, Feb. 2018, doi: 10.1016/j.jestch.2018.01.004.

[31] W. Gong, K. Zhang, C. Yang, M. Yi, and J. Wu, “Adaptive Visual Inspection Method for Transparent Label Defect Detection of Curved Glass Bottle,” in *Proceedings - 2020 International Conference on Computer Vision, Image and Deep Learning, CVIDL 2020*, Institute of Electrical and Electronics Engineers Inc., Jul. 2020, pp. 90–95. doi: 10.1109/CVIDL51233.2020.00024.

[32] F. Li, Z. Hang, G. Yu, G. Wei, and C. Xinyu, “The method for glass bottle defects detecting based on machine vision,” in *2017 29th Chinese Control And Decision Conference (CCDC)*, IEEE, May 2017.

[33] W. Koodtalang, T. Sangsuwan, and S. Sukanna, “Glass Bottle Bottom Inspection Based on Image Processing and Deep Learning,” in *2019 Research, Invention, and Innovation Congress (RI2C)*, IEEE, Dec. 2019, pp. 1–5. doi: 10.1109/RI2C48728.2019.8999883.
[34] L. Fu, S. Zhang, Y. Gong, and Q. Huang, “Medicine Glass Bottle Defect Detection Based on Machine Vision,” in *2019 Chinese Control And Decision Conference (CCDC)*, IEEE, Jun. 2019, pp. 5681–5685. doi: 10.1109/CCDC.2019.8832688.
[35] J. Lu and X. Zhang, “A defect detection method for glass bottle mouth based on area segmentation,” in *2021 International Conference on Image, Video Processing, and Artificial Intelligence*, Y. Zhang and D. Zhang, Eds., SPIE, Nov. 2021, p. 17. doi: 10.1117/12.2608396.
[36] A. M. Kadhim, “Course in Programming Visual Basic 6.0 Second year,” 2022, p. 30.
[37] M. Wu *et al.*, “Prediction of the remaining time and time interval of pebbles in pebble bed HTGRs aided by CNN via DEM datasets,” *Nuclear Engineering and Technology*, vol. 55, no. 1, pp. 339–352, Jan. 2023, doi: 10.1016/j.net.2022.09.019.
[38] M. M and S. P, “COVID-19 infection prediction from CT scan images of lungs using Iterative Convolution Neural Network model,” *Advances in Engineering Software*, vol. 173, p. 103214, Nov. 2022, doi: 10.1016/j.advengsoft.2022.103214.
[39] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
[40] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.
[41] T. Bezdan and N. Bačanin Džakula, “Convolutional Neural Network Layers and Architectures,” in *Proceedings of the International Scientific Conference - Sinteza 2019*, Novi Sad, Serbia: Singidunum University, 2019, pp. 445–451. doi: 10.15308/Sinteza-2019-445-451.
[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
[43] D. Bhatt *et al.*, “Cnn variants for computer vision: History, architecture, application, challenges and future scope,” *Electronics (Switzerland)*, vol. 10, no. 20. MDPI, Oct. 01, 2021. doi: 10.3390/electronics10202470.
[44] Purwono, A. Ma’arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, “Understanding of Convolutional Neural Network (CNN): A Review,” *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2022, doi: 10.31763/ijrcs.v2i4.888.
[45] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” *Neural Comput*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/neco_a_00990.
[46] N. K. Manaswi, “Understanding and Working with Keras,” in *Deep Learning with Applications Using Python*, Berkeley, CA: Apress, 2018, pp. 31–43. doi: 10.1007/978-1-4842-3516-4_2.
[47] F. J. J. Joseph, S. Nonsiri, and A. Monsakul, “Keras and TensorFlow: A Hands-On Experience,” 2021, pp. 85–111. doi: 10.1007/978-3-030-66519-7_4.

[48] A. Zelinsky, “Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf],” *IEEE Robot Autom Mag*, vol. 16, no. 3, pp. 100–100, Sep. 2009, doi: 10.1109/MRA.2009.933612.

[49] M. Kalbasi and H. Nikmehr, “Noise-Robust, Reconfigurable Canny Edge Detection and its Hardware Realization,” *IEEE Access*, vol. 8, pp. 39934–39945, 2020, doi: 10.1109/ACCESS.2020.2976860.

[50] D. D. Han, T. C. Zhang, and J. Zhang, “Research and Implementation of an Improved Canny Edge Detection Algorithm,” *Key Eng Mater*, vol. 572, pp. 566–569, Sep. 2013, doi: 10.4028/www.scientific.net/KEM.572.566.

Appendix A Arduino Coding

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
const int outputPin = 6;
const int motorPin = 3;
char command;

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{
  Serial.begin(9600);
  pinMode(motorPin, OUTPUT);
  pinMode(outputPin, OUTPUT);
  digitalWrite(motorPin, LOW);
  analogWrite(outputPin, 0);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("     WELCOME       ");
  delay(2000);
  lcd.setCursor(0, 0);
  lcd.print("AUTOMATIC BOTTLE ");
  lcd.setCursor(0, 1);
  lcd.print("   INSPECTION    ");
  delay(2000);
}

void loop()
{
  if (Serial.available() > 0)
  {
    command = Serial.read();
    if (command == 'A')
    {
      digitalWrite(motorPin, HIGH);
      //analogWrite(outputPin, 255);
      lcd.clear();
      lcd.setCursor(0, 1);
      lcd.print("CONVEYOR STARTED ");
      delay(2000);
      lcd.setCursor(0, 1);
      lcd.print("CONVEYOR RUNNING ");
      delay(2000);
```

```
    }
    if (command == 'B')
    {
      digitalWrite(motorPin, LOW);
      //analogWrite(outputPin, 0);
      lcd.clear();
      lcd.setCursor(0, 1);
      lcd.print("CONVEYOR STOPED    ");
      delay(2000);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("AUTOMATIC BOTTLE ");
      lcd.setCursor(0, 1);
      lcd.print("   INSPECTION   ");
    }
  }
}
```

Appendix B VB.NET Coding

FORM 1

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
Imports System.Windows.Forms.VisualStyles.VisualStyleElement.ToolBar


Public Class Form1
    '------------------------------------------------
    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal [text] As String) 'Added to prevent
threading errors during receiveing of data
    '------------------------------------------------
    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load

        myPort = IO.Ports.SerialPort.GetPortNames()
        ComboBox1.Items.AddRange(myPort)


        PictureBox2.Visible = False

        Button2.Enabled = False
        Button3.Enabled = False

    End Sub
    '------------------------------------------------
    Private Sub ComboBox1_Click(sender As System.Object, e As
System.EventArgs) Handles ComboBox1.Click
    End Sub
    '------------------------------------------------
    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs)
Handles Button1.Click
        SerialPort1.PortName = ComboBox1.Text
        SerialPort1.BaudRate = ComboBox2.Text
        SerialPort1.Open()
        Button1.Enabled = False
        Button2.Enabled = True
        Button3.Enabled = True
        Button4.Enabled = True

        ' show PictureBox7
        PictureBox7.Visible = False

        ' Set CheckBox states
        CheckBox1.Checked = True
        CheckBox2.Checked = False

    End Sub
    '------------------------------------------------
    Private Sub Button2_Click(sender As System.Object, e As System.EventArgs)
Handles Button2.Click
        SerialPort1.Write("A" & vbCr) 'concatenate with \n

        ' Show PictureBox2
```

```
        PictureBox2.Visible = True
        ' show PictureBox7
        PictureBox7.Visible = True
        ' show PictureBox8
        PictureBox8.Visible = False

    End Sub

    Private Sub Button4_Click(sender As System.Object, e As System.EventArgs)
Handles Button4.Click
        SerialPort1.Close()
        Button1.Enabled = True
        Button2.Enabled = False
        Button4.Enabled = False

        ' show PictureBox7
        PictureBox7.Visible = True
        ' Set CheckBox states
        CheckBox1.Checked = False
        CheckBox2.Checked = True

        Button2.Enabled = False
        Button3.Enabled = False

    End Sub


    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        SerialPort1.Write("B" & vbCr) 'concatenate with \n

        ' Hide PictureBox2
        PictureBox2.Visible = False
        ' show PictureBox7
        PictureBox7.Visible = False
        ' show PictureBox8
        PictureBox8.Visible = True
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Me.Hide()  ' Hide Form1
        Dim MyForm As New Form2
        MyForm.Show()  ' Show Form2
    End Sub


End Class
```

<u>FORM 2</u>

```
Imports System.Data.OleDb

Public Class Form2
    Dim connectionString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\lione\OneDrive\Desktop\prediction_Bottle\DB\bottle_db.mdb;"
    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
        'TODO: This line of code loads data into the
'Bottle_dbDataSet.bottleresult' table. You can move, or remove it, as needed.
        Dim sql As String = "SELECT * FROM bottleresult"
        Dim connection As New OleDbConnection(connectionString)
        Dim dataadapter As New OleDbDataAdapter(sql, connection)
        Dim ds As New DataSet()
        connection.Open()
        dataadapter.Fill(ds, "bottleresult")
        connection.Close()
        DataGridView1.DataSource = ds
        DataGridView1.DataMember = "bottleresult"

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim Div As String
        Div = "Proper"
        Me.BindGrid(Div)
    End Sub
    Private Sub BindGrid(ByVal Div As String)

        Using con As New OleDbConnection(ConnectionString)
            Using cmd As New OleDbCommand("SELECT * from bottleresult where
bottle_result='" & Div & "'", con)
                cmd.CommandType = CommandType.Text
                Using sda As New OleDbDataAdapter(cmd)
                    Using dt As New DataTable()
                        sda.Fill(dt)
                        DataGridView1.DataSource = dt
                    End Using
                End Using
            End Using
        End Using


    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim Div As String
        Div = "Defective"
        Me.BindGrid(Div)
    End Sub
    Private Sub DataGridView1_RowPrePaint(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.DataGridViewRowPrePaintEventArgs) Handles
DataGridView1.RowPrePaint

        If e.RowIndex >= 0 Then
            Me.DataGridView1.Rows(e.RowIndex).Cells(0).Value = e.RowIndex + 1
        End If
    End Sub
End Class
```

Appendix C CNN Prepossing Module Coding

```
import numpy as np
import os
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
DIRECTORY = r'C:\Users\lione\OneDrive\Desktop\prediction_Bottle\dataset'

CATEGORIES = ['Proper', 'Defective']
data = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        label = CATEGORIES.index(category)
        arr = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
        new_arr = cv2.resize(arr, (60, 60))
        data.append([new_arr, label])
data

[[array([[192, 193, 195, ..., 185, 182, 178],
         [191, 193, 196, ..., 184, 182, 181],
         [194, 195, 195, ..., 185, 183, 178],
         ...,
         [ 79, 106, 103, ...,  84,  76,  62],
         [ 41,  62,  64, ...,  15,  15,  16],
         [ 26,  37,  43, ...,  15,  15,  15]], dtype=uint8),
  1],
 [array([[191, 194, 195, ..., 183, 180, 179],
         [192, 194, 195, ..., 181, 180, 179],
         [192, 193, 194, ..., 182, 179, 179],
         ...,
         [ 63,  74,  98, ...,  92,  74,  47],
         [ 39,  59,  53, ...,  15,  15,  16],
         [ 25,  35,  40, ...,  13,  15,  15]], dtype=uint8),
  1],
 [array([[207, 210, 209, ..., 198, 195, 194],
         [206, 208, 208, ..., 196, 194, 191],
         [205, 207, 209, ..., 196, 193, 191],
         ...,
         [ 62,  62,  64, ...,  66,  65,  66],
         [ 59,  57,  59, ...,  69,  72,  69],
         [ 59,  61,  63, ...,  73,  62,  54]], dtype=uint8),
  0],
```

```
 [array([[215, 215, 218, ..., 195, 194, 191],
...
         [ 58,  64,  98, ...,  63,  42,  24],
         [ 38,  58,  51, ...,  13,  13,  15],
         [ 22,  34,  34, ...,  13,  13,  14]], dtype=uint8),
  0],
 ...]

data[0][1]

import random

random.shuffle(data)
X = []
y = []

for features, label in data:
    X.append(features)
    y.append(label)

X = np.array(X)
y = np.array(y)

X

array([[[184, 184, 185, ..., 191, 189, 186],
        [184, 184, 186, ..., 189, 188, 187],
        [181, 185, 186, ..., 190, 187, 184],
        ...,
        [ 64,  63,  89, ...,  90,  62,  44],
        [ 36,  56,  53, ...,  14,  14,  14],
        [ 21,  33,  34, ...,  12,  13,  14]],

       [[196, 198, 199, ..., 189, 187, 188],
        [197, 199, 198, ..., 189, 186, 186],
        [197, 198, 199, ..., 189, 186, 186],
        ...,
        [ 79, 108, 107, ...,  99,  90,  69],
        [ 43,  67,  66, ...,  18,  18,  19],
        [ 28,  40,  48, ...,  15,  17,  17]],

       [[189, 189, 191, ..., 179, 178, 173],
        [188, 190, 190, ..., 176, 175, 175],
        [188, 188, 190, ..., 177, 175, 174],
        ...,
        [ 60,  66,  96, ...,  66,  56,  26],
        [ 38,  58,  47, ...,  13,  12,  13],
        [ 22,  33,  34, ...,  11,  12,  12]],
```

```
       ...,
...
        [196, 197, 199, ..., 188, 184, 185],
        ...,
        [ 85, 104, 105, ..., 102,  99,  77],
        [ 45,  66,  68, ...,  17,  16,  18],
        [ 28,  38,  47, ...,  16,  16,  16]]], dtype=uint8)

y

array([1, 1, 0, ..., 1, 1, 1])

import pickle

pickle.dump(X, open('X.pkl', 'wb'))
pickle.dump(y, open('y.pkl', 'wb'))
```

Appendix D CNN Training and Classification Module Coding

```
import pickle

X = pickle.load(open('X.pkl', 'rb'))
y = pickle.load(open('y.pkl', 'rb'))

X = X/255

X = X.reshape(-1, 60, 60, 1)

from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten

from keras.callbacks import TensorBoard
import time

dense_layers = [3]
conv_layers = [3]
neurons = [64]


for dense_layer in dense_layers:
    for conv_layer in conv_layers:
        for neuron in neurons:

          NAME = '{}-denselayer-{}-convlayer-{}-neuron-
{}'.format(dense_layer, conv_layer, neuron, int(time.time()))
          tensorboard = TensorBoard(log_dir = 'logs2\\{}'.format(NAME))


          model = Sequential()

          for l in range(conv_layer):
             model.add(Conv2D(neuron, (3,3), activation = 'relu'))
             model.add(MaxPooling2D((2,2)))

          model.add(Flatten())

          model.add(Dense(neuron, input_shape = X.shape[1:], activation =
'relu'))

          for l in range(dense_layer - 1):
             model.add(Dense(neuron, activation = 'relu'))

          model.add(Dense(2, activation = 'softmax'))

          model.compile(optimizer='adam',
                        loss='sparse_categorical_crossentropy',
                        metrics=['accuracy'])


print('==================================================================
==================================================================')

print('==================================================================
==================================================================')
          print('=========================================== RUNNING
```

```
MODEL
==========================================================================
')
          print('==================================================='+
NAME +
'=========================================================================')

print('=======================================================================
==================================================================')

print('=======================================================================
==================================================================')

          model.fit(X, y, epochs=100, batch_size = 32,
validation_split=0.1, callbacks = [tensorboard])

          model.save('3x3x64-properVSdefect.model')
```

Appendix E CNN Real Time Detection Module

```
# import the necessary packages
import numpy as np
import cv2
import pyodbc
import pandas as pd
import keras
from datetime import datetime
import pickle
import pandas as pd
from sklearn import svm
from sklearn.model_selection import GridSearchCV
import os
import matplotlib.pyplot as plt
from skimage.transform import resize
from skimage.io import imread
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import
classification_report,accuracy_score,confusion_matrix
import pickle
import time
Categories=['Defective','Proper']

now = datetime.now()
formatted_date = now.strftime('%Y-%m-%d %H:%M:%S')
try:
    con_string = r'DRIVER={Microsoft Access Driver (*.mdb,
*.accdb)};DBQ=C:\Users\lione\OneDrive\Desktop\prediction_Bottle\DB\bottle
_db.mdb;'
    conn = pyodbc.connect(con_string)
    print("Connected To Database")

except pyodbc.Error as e:
    print("Error in Connection", e)
# initialize the HOG descriptor/person detector
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

cv2.startWindowThread()
index=["color", "color_name", "hex", "R", "G", "B"]
csv = pd.read_csv('colors.csv', names=index, header=None)
def recognize_color(R,G,B):
    minimum = 10000
    for i in range(len(csv)):
        d = abs(R- int(csv.loc[i,"R"])) + abs(G- int(csv.loc[i,"G"]))+
abs(B- int(csv.loc[i,"B"]))
        if(d<=minimum):
            minimum = d
            cname = csv.loc[i,"color_name"]
    return cname
CATEGORIES = ['Proper', 'Defective']


def image(path):
    img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
    new_arr = cv2.resize(img, (60, 60))
    new_arr = np.array(new_arr)
```

```
    new_arr = new_arr.reshape(-1, 60, 60, 1)
    return new_arr
# open webcam video stream
cap = cv2.VideoCapture(1)
model = keras.models.load_model('3x3x64-properVSdefect.model')
x = 0

while(True):
    # Capture frame-by-frame
    time.sleep(0.1)
    ret, frame = cap.read()
    cv2.imwrite('detect.jpg', frame)
    h, w, c = frame.shape
    ar = h/w
    h=int(h/2)
    w=int(w/2)
    b,g,r = frame[h,w]
    b = int(b)
    g = int(g)
    r = int(r)
    text=''
    cv2.imshow('frame',frame)
    img_resize=resize(frame,(150,150))
    img_gray = cv2.cvtColor((img_resize * 255).astype(np.uint8),
cv2.COLOR_RGB2GRAY)
    ret, thresh1 = cv2.threshold(img_gray, 120, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)
    # Find Canny edges
    edged = cv2.Canny(frame, 30, 200)

    # Finding Contours
    # Use a copy of the image e.g. edged.copy()
    # since findContours alters the image
    contours, hierarchy = cv2.findContours(edged,
        cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    print("Number of Contours found = " + str(len(contours)))

    if(len(contours)>110):
        print("started done")
        x += 1
        print(x)
        if(x>5):
            print("counting done")
            x=0
            prediction = model.predict([image('detect.jpg')])
            print(prediction)
            print(CATEGORIES[prediction.argmax()])
            try:
                text = recognize_color(r,g,b)
                print(text)
            except Exception:
                print("No Colour")
                print("No Aspect Ration")
            cursor = conn.cursor()
            cursor.execute('insert into
bottleresult(date_t,bottle_result,bottle_colour) values(?,?,?)',
(formatted_date,CATEGORIES[prediction.argmax()],text))
            conn.commit()
            print('Data Inserted')
            time.sleep(5)
        time.sleep(1)
```

```
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
# and release the output
#out.release()
# finally, close the window
cv2.destroyAllWindows()
cv2.waitKey(1)
```