Volume 2, Issue 4, Apr 2026 https://doi.org/10.5281/zenodo.19731004 ISSN : 3139-1478

PNEUMATIC CONVEYOR SORTING SYSTEM WITH MACHINE VISION

[1]K.Satheeesh Kumar, [2]S.Samson, [3]J.Rahul, [4]A.Goutham, [5]Ganapathy

[1]Assistant Professor, [2345]Students
[12345]Mechanical engineering
[12345]Salem College of Engineering and Technology, Salem- Attur Main Road, M.Perumapalayam, Selliamman Nagar, Salem.
[1]skysatheesh@gmail.com, [2]samsons280405@gmail.com, [3]rahulkalamani@gmail.com,
[4]gourigourisri@gmail.com

Abstract—**Automation in industrial sorting systems plays a vital role in improving efficiency, accuracy, and productivity. This project presents the design and implementation of a Pneumatic Conveyor Sorting System integrated with Machine Vision for automated object classification based on size and shape. The system consists of a proximity sensor and PLC-based logic for size identification, along with a camera module for shape detection. A Raspberry Pi acts as the central processing unit, receiving inputs from both the sensor system and the vision module.The captured data is processed using image processing techniques to extract object features such as dimensions and contours. Based on the classification results, control signals are sent through a relay module to actuate servo motors and pneumatic actuators. These actuators direct objects into designated bins according to their size and shape.The integration of sensor-based size detection with vision-based shape identification ensures high accuracy and reliability. The system reduces manual effort, minimizes sorting errors, and enhances operational efficiency. This solution is suitable for applications in manufacturing, packaging, and material handling industries, with future scope for advanced AI-based classification and IoT integration.**

Index Terms—**Pneumatic Sorting System, Machine Vision, Size Identification, Shape Detection, Raspberry Pi, Image Processing, Conveyor Automation, Industrial Automation.**

I. Introduction

Industrial automation involves the application of control systems, machinery, and information technologies to operate industrial processes with minimal human intervention. Over recent decades, automation has significantly transformed manufacturing by improving productivity, consistency, accuracy, and operational efficiency. Industries such as automotive, food processing, packaging, electronics, and logistics increasingly rely on automation to meet rising market demands and remain competitive.

The primary objective of industrial automation is to ensure consistent product quality while reducing dependence on manual labor. Automated systems enable continuous operation, minimize human error, and enhance process reliability. With advancements in embedded systems, sensors, and software-based control, automation has evolved to include intelligent decision-making capabilities, allowing industries to manage complex operations efficiently.

II. Problem Identification

In many small- and medium-scale industries, product sorting is still performed manually or using semi-automated systems. These methods are labor-intensive, time-consuming, and prone to human error and fatigue, resulting in inconsistent accuracy and reduced productivity. Manual sorting is unsuitable for high-speed and continuous production environments.

Although pneumatic systems provide fast actuation, conventional pneumatic sorting systems lack intelligent control, leading to unnecessary actuation, air leakage, energy wastage, and reduced efficiency. There is a need for an intelligent, accurate, and energy-efficient sorting system that overcomes these limitations while remaining affordable.

III. Existing system

Prolonged working hours, operator fatigue significantly reduces sorting accuracy, leading to inconsistent output quality and increased rejection rates. Additionally, manual sorting limits production speed and requires a large workforce, resulting in higher labor costs and reduced overall productivity.

To overcome the drawbacks of purely manual operations, many industries have adopted semi-automated sorting systems. In such systems, conveyors and basic mechanical aids are introduced to assist material handling, while decision-making and supervision remain dependent on human operators. Sensors such as limit switches or basic photoelectric sensors may be used to detect object presence, but the final classification often relies on manual judgment. Although semi-automated systems offer improved throughput compared to manual sorting, they still suffer from inconsistency and lack the intelligence required to handle variations in object characteristics. Continuous operator involvement increases the probability of errors and restricts system efficiency, especially in high-volume production environments.

Conventional automated sorting systems represent a further advancement by utilizing sensors and actuators to reduce human involvement. These systems typically employ sensors such as infrared sensors, inductive sensors, capacitive sensors, proximity sensors, and load cells to detect specific physical parameters of objects. Based on sensor signals, mechanical or pneumatic actuators divert objects into predefined bins. While these systems significantly improve speed and reduce labor dependency, they are limited in scope and flexibility. Sensor-based systems can detect only predefined parameters and fail to accurately classify objects with irregular shapes, overlapping dimensions, or surface variations. Any change in product specification requires physical repositioning of sensors or mechanical reconfiguration, leading to system downtime and increased maintenance effort.

Pneumatic-based sorting systems are widely used in existing industrial setups due to their fast response time, simple construction, and suitability for repetitive operations. Pneumatic cylinders controlled by solenoid valves are commonly used to push or divert objects from conveyor belts into sorting bins. Although pneumatic actuation offers advantages such as reliability and ease of maintenance, existing pneumatic sorting systems often operate using basic on–off control logic without intelligent decision-making. This results in unnecessary actuator operation, excessive compressed air consumption, and reduced energy efficiency. Air leakage, pressure fluctuations, and improper timing further affect sorting precision, especially at higher conveyor speeds.

One of the major limitations of existing sorting systems is their inability to achieve high accuracy and consistency. Manual and sensor-based systems struggle to detect subtle differences in size and shape, leading to frequent misclassification. Mechanical tolerances, sensor inaccuracies, and environmental factors such as lighting and dust further degrade system performance. In industries with strict quality requirements, such inaccuracies result in defective products, customer dissatisfaction, and regulatory non-compliance. Existing systems lack advanced perception capabilities to ensure reliable and repeatable sorting performance.

Another significant drawback of existing systems is the lack of flexibility and adaptability. Most traditional sorting systems are designed for fixed product dimensions and predefined operating conditions. When production requirements change or new product variants are introduced, extensive mechanical modifications or sensor replacements are required. This lack of adaptability increases downtime and limits the system's ability to respond to dynamic manufacturing needs. As modern industries demand rapid reconfiguration and scalability, existing systems fail to provide the required level of flexibility.

Operational and maintenance costs also pose a major challenge in existing sorting systems. Manual systems require continuous labor investment, while semi-automated and automated systems involve frequent maintenance due to mechanical wear and sensor recalibration. Pneumatic systems, in particular, consume significant energy when air usage is not optimized. Continuous compressor operation, air leakage, and inefficient actuation increase energy consumption and operational expenses. These factors reduce the economic viability of existing systems, especially for small- and medium-scale industries.

Furthermore, most existing sorting systems lack real-time monitoring and feedback mechanisms. Fault detection is often manual, resulting in delayed maintenance and unexpected system failures. The absence of performance monitoring, data logging, and diagnostic features prevents process optimization and predictive maintenance. Without intelligent monitoring, system reliability and productivity are compromised, leading to increased downtime and reduced operational efficiency.

Overall, the existing sorting systems used in industries are characterized by high dependency on manual labor, limited accuracy, poor adaptability, inefficient pneumatic control, and increased operational costs. These limitations highlight the need for an advanced automated sorting solution that integrates intelligent perception, precise control, and efficient actuation. The shortcomings of existing systems form the basis for proposing a Pneumatic Conveyor Sorting System integrated with Machine Vision, aimed at overcoming these challenges and meeting modern industrial requirements.

IV. Proposed System

The proposed system introduces an automated Pneumatic Conveyor Sorting System integrated with Machine Vision to overcome the limitations of conventional and manual sorting methods. The system is designed to accurately sort objects based on physical characteristics such as size and shape by combining mechanical material handling, intelligent sensing, image processing, embedded control, and pneumatic actuation. The primary objective of the proposed system is to achieve high sorting accuracy, reduced labor dependency, improved efficiency, and adaptability while maintaining cost-effectiveness suitable for small- and medium-scale industries.

In the proposed system, a conveyor belt mechanism is employed as the primary material handling unit. The conveyor ensures smooth and continuous transportation of objects through the detection and sorting zones. The belt speed is carefully controlled to provide sufficient time for sensor triggering, image acquisition, and processing without compromising throughput. Mechanical stability and proper alignment of rollers and belt tension are ensured to minimize vibration and object displacement, which are critical for accurate sensing and image capture.

1. Features Of Proposed System

A. The Future of Smart Parking Systems: Modern urban infrastructure increasingly depends on smart parking systems to streamline traffic flow, 18 optimize space utilization, and enhance user convenience. These intelligent parking solutions rely on the seamless integration of hardware, software, communication protocols, and AI-driven analytics to create efficient and responsive environments. As technology advances, smart parking systems are poised to become even more sophisticated, sustainable, and highly adaptive to real-world urban challenges.

B. Extended Hardware Integrate: While Arduino UNO has played a pivotal role in early smart parking prototypes, the future landscape is shifting toward more advanced microcontrollers and embedded systems. The adoption of ESP32, STM32, and Raspberry Pi enhances functionality, introducing built-in Wi-Fi and Bluetooth support, higher processing power, and expanded memory for real-time analytics. Edge Computing. By integrating AI-capable chips such as Google's Coral Edge TPU, Nvidia Jetson, or Intel Movidius, smart parking systems will process data at the edge rather than relying entirely on cloud services. This reduces latency and enables real-time decision-making..Advanced parking systems will transition toward LiDAR sensors, HD cameras, and computer vision models for precise vehicle tracking, license plate recognition, and occupancy detection.

C. **Enhanced Multi-Sensor Framework & Intelligent Space Allocation:** The conventional use of IR Sensors and Ultrasonic Sensors for vehicle detection is evolving into multi-sensor frameworks that combine radar technology, RFID tracking, and AI-powered predictive monitoring.AI models trained on historical parking trends can forecast congestion, predict space availability, and optimize slot distribution dynamically . RFID tags embedded in registered vehicles can facilitate automated entry and exit tracking, ensuring a seamless parking experience without manual intervention. Smart parking systems will communicate directly with 19 connected vehicles, allowing real-time slot reservations and navigation guidance to available parking spaces.

D. **Intelligent Payment and Block chain Integration:** Implementing smart contracts ensures transparent, tamper-proof automated payments for parking services. Future parking systems may support crypto-based transactions, QR payments, and biometric authentication for instant, contactless payments.

E. **Cloud-Connected Predictive Analytics & Sustainability Practices:** Green Energy Integration Smart parking stations will increasingly utilize solar powered hubs, energy-efficient lighting, and AI-driven carbon footprint reduction strategies . EV Charging-Embedded Parking Systems with the rise of electric vehicles (EVs), smart parking systems will integrate wireless EV charging docks, ensuring a seamless experience for eco-conscious drivers. The continuous transformation of smart parking technology is key to shaping future-ready cities, reducing congestion, and promoting sustainable urban mobility. As AI, IoT, and automation synergize with cloud infrastructure and block chain-based solutions, the next generation of parking systems will redefine convenience, efficiency, and environmental consciousness. The journey toward AI-powered, IoT-enhanced smart parking is only beginning. As these systems evolve, they will revolutionize urban landscapes, improve traffic flow, and create intelligent, interconnected mobility ecosystems.

2. Objectives oF Proposed System

The primary objective of this project is to design and develop an automated pneumatic conveyor sorting system integrated with machine vision to sort objects based on size and shape. The system aims to achieve accurate, non-contact detection, intelligent decision-making using Raspberry Pi, and precise pneumatic actuation. Additional objectives include reducing manual labor, minimizing operational costs, improving energy efficiency, and ensuring flexibility and scalability for future enhancements.

V. Methodology

The Pneumatic Conveyor Sorting System is developed by first selecting suitable components including a conveyor belt, proximity sensor, camera module, Raspberry Pi, solenoid valves, pneumatic actuators, and relay module.

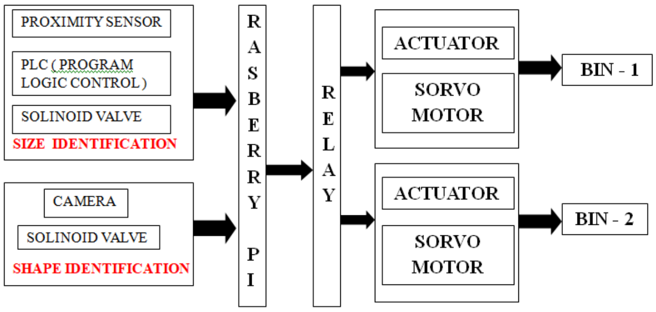

Fig – 1 - Methodology

A. Selection of Components:

The first step involves selecting the required hardware and software components necessary for implementing the pneumatic conveyor sorting system. The major components include a conveyor belt setup, proximity sensor, camera module, Raspberry Pi, relay module, solenoid valves, pneumatic cylinders, air compressor, and power supply units. Software tools such as Python, OpenCV, and Raspberry Pi OS are selected for image processing and system control.

B. Design of System Architecture:

In this stage, the overall system architecture is designed based on the block diagram. The layout includes proper positioning of the conveyor belt, sensor placement for size detection, camera mounting for shape identification, and actuator alignment for accurate sorting into bins. This step ensures efficient coordination between sensing, processing, and actuation units.

C. Development of Conveyor and Pneumatic Setup:

The conveyor belt mechanism is assembled to transport objects smoothly through the detection and sorting zones. Pneumatic cylinders and solenoid valves are installed at appropriate locations along the conveyor to divert objects into Bin-1 and Bin-2. Proper air pressure regulation is ensured to achieve consistent actuator performance.

D. Installation of Proximity Sensor for Size Identification:

A proximity sensor is mounted above the conveyor belt to detect the presence and size of objects. As objects pass through the sensing zone, the sensor provides size-related input signals to the Raspberry Pi. This enables preliminary classification of objects based on their height or size threshold.

E. Camera Installation for Shape Identification:

A camera module is installed above the conveyor belt to capture real-time images of moving objects. Controlled lighting conditions are maintained to ensure clear image acquisition. The captured images are used for shape detection and classification through machine vision techniques.

F. Programming of Raspberry Pi:

The Raspberry Pi is programmed using Python to process inputs from the proximity sensor and camera module. OpenCV libraries are used for image preprocessing, contour detection, and shape recognition. Based on size and shape analysis, decision logic is implemented to determine the appropriate sorting action.

G. Relay and Solenoid Valve Integration:

A relay module is interfaced between the Raspberry Pi and the solenoid valves to provide electrical isolation and safe switching. Based on the control signals generated by the Raspberry Pi, the corresponding solenoid valve is activated to control the pneumatic actuator for sorting.

H. Actuation and Sorting Operation:

When the classified object reaches the sorting point, the pneumatic actuator is triggered at the correct time to divert the object into the designated bin. Bin-1 is used for one category of objects, while Bin-2 is used for another category, as defined by size and shape identification logic.

I. Individual Component Testing:

Each subsystem, including sensors, camera, Raspberry Pi, relay module, and pneumatic actuators, is tested individually to verify proper functionality. This step helps in identifying and correcting errors before full system integration.

J. System Integration and Testing:

All hardware and software components are integrated, and the complete system is tested under real-time operating conditions. Synchronization between conveyor movement, sensing, processing, and actuation is verified to ensure accurate sorting.

K. Validation of Sorting System:

The final step involves validating the system by testing it with objects of different sizes and shapes. Sorting accuracy, response time, and reliability are evaluated to ensure that the system meets the design and functional requirements.

VI. MODELING AND ANALYSIS

The modeling and analysis of the proposed Pneumatic Conveyor Sorting System focus on mechanical design, image processing performance, and pneumatic response characteristics. The system is modeled as an integrated automation unit consisting of a conveyor mechanism, vision-based detection module, and pneumatic actuation system. Mathematical and functional modeling techniques are used to analyze object detection accuracy, actuator response time, and overall system efficiency. Performance parameters such as conveyor speed, image processing delay, air pressure requirements, and sorting accuracy are evaluated to ensure optimal operation.

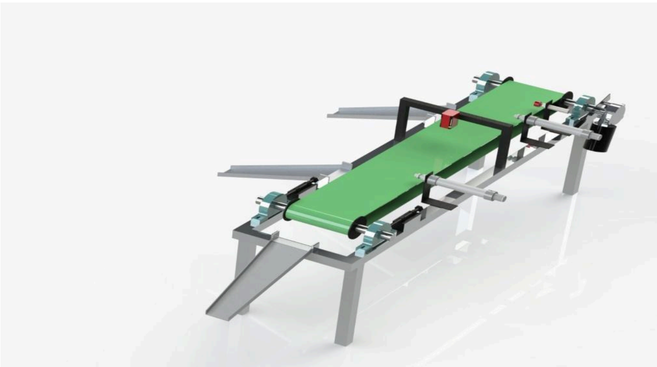

Fig -2 - 3D view of Pneumatic conveyor sorting system with machine vision.

VII. ARCHITECTURE OF THE PNEUMATIC CONVEYOR SORTING SYSTEM WITH MACHINE VISION

The Pneumatic Conveyor Sorting System integrated with Machine Vision is designed with a structured and modular architecture to achieve automated and accurate sorting based on object size and shape. The system combines mechanical handling, intelligent sensing, embedded processing, and pneumatic actuation to ensure efficient and reliable operation. The overall architecture enables seamless interaction between hardware and software components while maintaining flexibility and scalability.

The conveyor belt mechanism forms the core of the system, responsible for transporting objects through the sensing and sorting zones. Objects placed on the conveyor move at a controlled speed to allow sufficient time for detection and classification. A proximity sensor is installed above the conveyor to detect object presence and identify size-related characteristics. This sensor provides real-time input to the control unit, enabling preliminary classification based on object dimensions.

A camera module mounted above the conveyor captures images of objects for shape identification. The captured images are processed using image processing algorithms implemented through OpenCV. This machine vision approach enables accurate and non-contact shape detection, overcoming the limitations of traditional sensor-based methods. Controlled lighting conditions are maintained to ensure consistent image quality and reliable classification.

The Pneumatic Conveyor Sorting System integrated with Machine Vision is designed with a structured and modular architecture to achieve automated and accurate sorting based on object size and shape. The system combines mechanical handling, intelligent sensing, embedded processing, and pneumatic actuation to ensure efficient and reliable operation. The overall architecture enables seamless interaction between hardware and software components while maintaining flexibility and scalability.

The conveyor belt mechanism forms the core of the system, responsible for transporting objects through the sensing and sorting zones. Objects placed on the conveyor move at a controlled speed to allow sufficient time for detection and classification. A proximity sensor is installed above the conveyor to detect object presence and identify size-related characteristics. This sensor provides real-time input to the control unit, enabling preliminary classification based on object dimensions.

A camera module mounted above the conveyor captures images of objects for shape identification. The captured images are processed using image processing algorithms implemented through OpenCV. This machine vision approach enables accurate and non-contact shape detection, overcoming the limitations of traditional sensor-based methods. Controlled lighting conditions are maintained to ensure consistent image quality and reliable classification.

VIII. BLOCK DIAGRAM

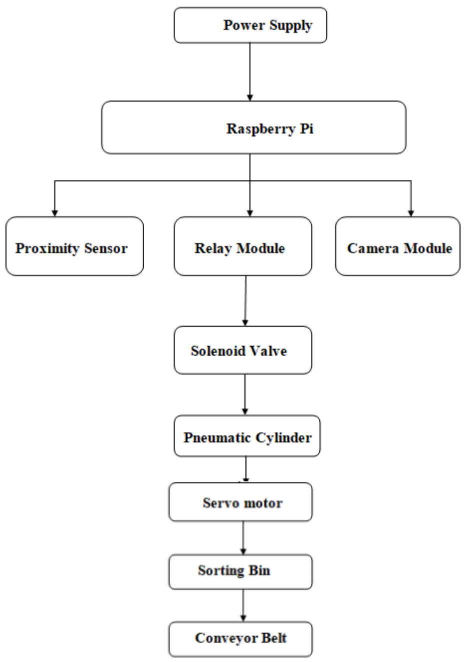

Fig – 3- Block diagram

- Before starting the system operation, ensure that the Raspberry Pi operating system is properly installed and the camera interface is enabled. Verify that the image processing libraries such as OpenCV are correctly configured.
- Connect the power supply to the Raspberry Pi, conveyor motor, relay module, and pneumatic solenoid valve, ensuring appropriate voltage levels are maintained for each component.
- Install the proximity sensor near the conveyor belt and connect its VCC pin to the power supply, GND pin to common ground, and output pin to a GPIO pin of the Raspberry Pi for size detection.
- Mount the camera module above the conveyor belt and connect it to the Raspberry Pi using the CSI or USB interface to capture real-time images of the moving objects.
- Interface the relay module with the Raspberry Pi by connecting the input pin of the relay to a GPIO pin and the ground and VCC pins to the corresponding power terminals.
- Connect the solenoid valve to the relay module so that the valve can be switched ON and OFF based on control signals generated by the Raspberry Pi.
- Connect the pneumatic cylinder to the solenoid valve and air compressor to enable mechanical actuation for object sorting.
- Ensure the conveyor belt motor is properly connected and synchronized with the sensing and sorting operations.
- Based on the size and shape analysis performed by the Raspberry Pi, control signals are generated to activate the relay and solenoid valve at the correct time.
- The pneumatic cylinder diverts the detected object into the appropriate bin, completing the automated sorting process.

IX. COMPONENT DESCRIPTION

A. Conveyor Belt Mechanism:

The conveyor belt mechanism serves as the primary material handling unit of the sorting system. It is responsible for transporting objects from the loading point through the sensing and sorting zones. The conveyor belt operates at a controlled speed to ensure accurate sensing and actuation. Proper alignment of rollers and belt tension ensures smooth and continuous movement of objects, minimizing vibration and positional errors during sorting.

B. Raspberry Pi:

The Raspberry Pi acts as the central processing and control unit of the pneumatic conveyor sorting system. It receives input signals from the proximity sensor and image data from the camera module. Using Python programming and OpenCV libraries, the Raspberry Pi processes visual information to identify object shape and size. Based on the processed data, it generates control signals to actuate the pneumatic system. The Raspberry Pi enables real-time decision-making and flexible system configuration.

C. Proximity Sensor:

The proximity sensor is used to detect the presence and relative size of objects on the conveyor belt. It operates on a non-contact sensing principle, ensuring reliable detection without physical wear. When an object passes through the sensing zone, the sensor generates a signal that is sent to the Raspberry Pi. This input is used for preliminary size classification and timing synchronization for sorting.

D. Camera Module:

The camera module captures real-time images of objects moving on the conveyor belt. These images are used for shape detection and classification through machine vision algorithms. The camera is mounted at a fixed position to ensure consistent image acquisition. Proper lighting conditions are maintained to enhance image clarity and improve the accuracy of image processing results.

E. Relay Module:

The relay module acts as an interface between the Raspberry Pi and high-power pneumatic components. It provides electrical isolation and protects the Raspberry Pi from voltage fluctuations. Based on control signals from the Raspberry Pi, the relay switches the solenoid valve ON or OFF to initiate pneumatic actuation.

F. Solenoid Valve:

The solenoid valve controls the flow of compressed air to the pneumatic cylinder. When energized through the relay module, the solenoid valve opens or closes air passages, enabling precise control of pneumatic actuation. This component plays a crucial role in converting electrical signals into mechanical motion.

G. Pneumatic Cylinder:

The pneumatic cylinder performs the mechanical sorting operation by diverting objects into designated bins. It converts compressed air energy into linear motion, providing fast and reliable actuation. Pneumatic cylinders are chosen for their simplicity, durability, and suitability for repetitive industrial operations.

H. Sorting Bins:

Sorting bins are used to collect objects based on their classified size and shape. Each bin is positioned strategically to receive objects diverted by the pneumatic actuator. This arrangement ensures efficient separation and easy retrieval of sorted items.

X. CIRCUIT DIAGRAM OF PNEUMATIC CONVEYOR SORTING SYSTEM

The circuit diagram of the Pneumatic Conveyor Sorting System illustrates the electrical and control interconnections between sensors, control units, and actuators used for automated sorting. It shows how input devices such as proximity sensors and the camera module provide signals to the Raspberry Pi and PLC for size and shape identification. The control signals are processed and routed through relay modules to operate solenoid valves accurately.

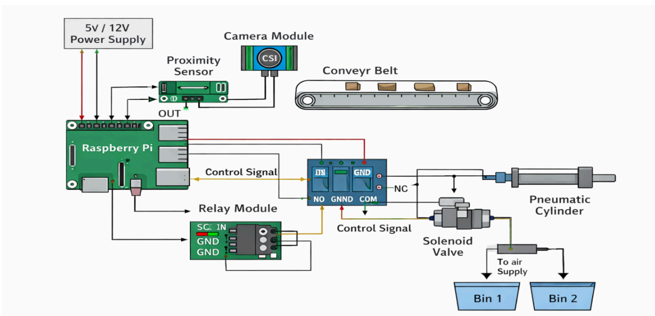

Fig – 4 - Circuit Diagram

These solenoid valves regulate the pneumatic actuators responsible for diverting objects into designated bins. Overall, the circuit diagram represents a synchronized control structure that ensures precise, reliable, and efficient sorting operations.

XI. EXPLANATION OF SYSTEM OPERATION

The **Pneumatic Conveyor Sorting System** operates in a sequential and coordinated manner to achieve accurate object sorting based on size and shape, as described below:

A. System Initialization:

When the system is powered ON, the Raspberry Pi initializes all connected hardware components, including the conveyor motor, proximity sensor, camera module, relay unit, and pneumatic solenoid valves. The machine vision libraries and control logic are also loaded to prepare the system for real-time operation.

B. Object Detection on Conveyor:

As objects move along the conveyor belt, the proximity sensor detects the presence of an object at the inspection zone. This detection signal is sent to the Raspberry Pi, triggering the image acquisition and size evaluation process.

C. Size Identification Process:

The size identification module, assisted by sensor input and PLC logic, determines whether the object meets predefined size thresholds. Based on this evaluation, the Raspberry Pi categorizes the object and prepares the corresponding control action.

D. Shape Identification Using Machine Vision:

Simultaneously, the camera captures images of the moving object. Using OpenCV-based image processing algorithms, the Raspberry Pi analyzes the object’s shape by extracting features such as contours and dimensions, enabling accurate classification.

E. Data Processing and Decision Making:

The Raspberry Pi integrates the results from both size and shape identification modules. Based on the combined analysis, a sorting decision is made, and the appropriate output signal is generated.

F. Pneumatic Actuation and Sorting:

The control signal is sent through a relay module to the corresponding solenoid valve. This valve activates the pneumatic actuator, which mechanically diverts the object into the designated bin (Bin-1 or Bin-2) without interrupting conveyor motion.

G. Continuous Operation:

The system continuously repeats this cycle for each object, ensuring high-speed, accurate, and reliable sorting. By integrating machine vision with pneumatic control, the system minimizes human intervention while maintaining efficiency and consistency in industrial sorting applications.

XII. DESIGN OF PNEUMATIC SYSTEM

a **pneumatic actuation system** used in an **automated conveyor-based sorting application**. The system consists of two identical pneumatic actuator units supplied from a common compressed air source, enabling controlled and synchronized object sorting.

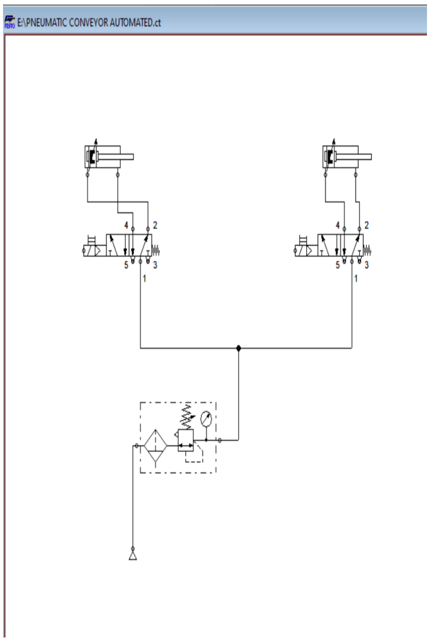

Fig. – 5 - Connection Diagram

A. **Compressed Air Source:** The compressed air source acts as the primary power supply for the pneumatic conveyor sorting system. It provides pressurized air required to operate the pneumatic cylinders and control valves. The air is generated using an external compressor and supplied continuously to ensure smooth and uninterrupted system operation.
B. **Air Preparation Unit (FRL Unit):** The air preparation unit consists of a filter, pressure regulator, and lubricator (FRL). The filter removes dust and moisture from the compressed air, while the regulator maintains the required operating pressure. This unit ensures clean, regulated air supply, which enhances the efficiency and lifespan of pneumatic components.
C. **Directional Control Valves (5/2 Solenoid Valves):** The system uses 5/2 directional control solenoid valves to control the movement of the pneumatic cylinders. These valves regulate the direction of airflow to extend or retract the cylinders. Electrical signals from the control system activate the solenoid valves, enabling precise and rapid actuation during the sorting process.
D. **Pneumatic Cylinders (Actuators):** Double-acting pneumatic cylinders are used as actuators to physically divert objects into their respective bins. When pressurized air is supplied to one side of the cylinder, the piston extends; when air is supplied to the opposite side, the piston retracts. This controlled motion allows accurate and efficient sorting of objects based on size or shape.
E. **Control Logic System:** The control logic system, implemented using a PLC or Raspberry Pi, processes input signals from sensors and machine vision modules. Based on the classification results, it generates appropriate output signals to activate the corresponding solenoid valves. This automated control ensures accurate, fast, and consistent sorting operations.
F. **Conveyor Integration:** The pneumatic actuators are integrated with the conveyor mechanism to enable real-time sorting without interrupting object flow. As objects move along the conveyor belt, the actuators respond instantly to control signals, diverting items smoothly into designated bins. This integration improves throughput and operational efficiency.
G. **Overall System Operation:** The combined operation of the air supply unit, control valves, actuators, and control logic ensures reliable and automated sorting. The system achieves fast

response times, reduced manual intervention, and high sorting accuracy, making it suitable for industrial applications such as packaging, food processing, and e-waste management.

XIII. DESIGN OF PLC LADER PROGRAM

The ladder diagram represents the control logic of the pneumatic conveyor sorting system using PLC programming. Input signals from sensors activate internal memory bits and timers to control system operation. ON-delay and OFF-delay timers ensure precise sequencing of pneumatic actuators through solenoid valves. This logic enables accurate, safe, and automated sorting of objects into designated bins.

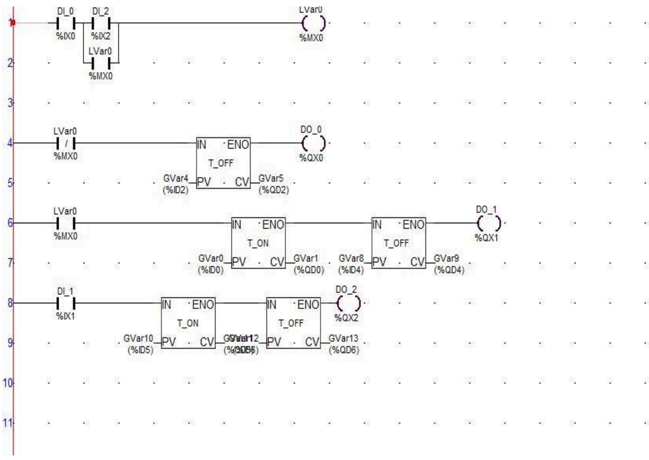

Fig. – 6 - PLC Ladder Diagram

A. nput Signal Initialization (DI_0 and DI_2): The first rung of the ladder diagram consists of two digital input contacts, DI_0 (%IX0) and DI_2 (%IX2), connected in series. These inputs act as start or condition signals for the system. When both inputs are energized, the internal control variable LVar0 (%MX0) is set. This rung establishes the basic enabling logic for the automated pneumatic conveyor sorting system.

B. Internal Control Variable (LVar0 – Memory Bit): The internal relay LVar0 (%MX0) functions as a memory or latch bit. Once activated, it maintains the system in an active state even if momentary input signals change. This internal variable is used repeatedly in subsequent rungs to control timers and outputs, ensuring synchronized operation across the sorting sequence.

C. Timer-Controlled Output DO_0 (T_OFF Operation): In the next rung, LVar0 (%MX0) triggers an OFF-delay timer (T_OFF). The preset value (PV) is provided by GVar4 (%ID2), and the current value (CV) is stored in GVar5 (%QD2). After the delay expires, the output DO_0 (%QX0) is activated. This output typically controls a solenoid valve or actuator, allowing delayed pneumatic movement for accurate object positioning.

D. Sequential Timing Control for DO_1 (T_ON and T_OFF): This rung uses a combination of ON-delay (T_ON) and OFF-delay (T_OFF) timers. When LVar0 is active, the T_ON timer starts with its preset from GVar0 (%ID0) and output stored in GVar1 (%QD0). After the ON delay, a second T_OFF timer operates using GVar8 (%ID4) and GVar9 (%QD4). This timed sequence finally activates DO_1 (%QX1), ensuring controlled extension and retraction of a pneumatic actuator.

E. Input-Based Actuation Using DI_1: The eighth rung is initiated by DI_1 (%IX1), which acts as an additional sensor or condition input, possibly from a proximity or limit switch. This input triggers another pair of timers, allowing the system to respond dynamically based on object detection or position confirmation.

F. Timed Control of Output DO_2: Upon activation of DI_1, the T_ON timer with preset GVar10 (%ID5) starts, followed by a T_OFF timer using GVar12 (%QD5) and GVar13 (%QD6). After completing the timing sequence, the output DO_2 (%QX2) is energized. This output may control a second pneumatic actuator used for directing objects into an alternate bin.

G. Output Devices and Pneumatic Actuation: The outputs DO_0, DO_1, and DO_2 represent digital outputs connected to solenoid valves or relays. These outputs translate PLC logic into physical pneumatic actions such as cylinder extension, retraction, or diversion. The use of timers ensures smooth, collision-free, and synchronized pneumatic motion.

XIV. PYTHON PROGRAM FOR MACHINE VISION-BASED PNEUMATIC CONVEYOR SORTING SYSTEM USING RASPBERRY PI

A. Initialization and Actuator Control Code for Pneumatic Conveyor Sorting System

This code segment initializes the core components required for controlling the pneumatic sorting system using a Raspberry Pi. It begins by importing essential libraries such as OpenCV for image processing, NumPy for numerical operations, RPi.GPIO for hardware interfacing, and time for delay control. The GPIO pins 17 and 27 are assigned to control relays connected to Bin 1 and Bin 2, respectively, and are configured as output pins using BCM mode. Both pins are initially set to LOW to ensure that the relays and actuators remain off at startup. The camera is then initialized using cv2.VideoCapture(0) to capture real-time images from the connected camera module. A function named activate_bin() is defined to control the sorting mechanism; based on the input parameter, it activates the corresponding relay by sending a HIGH signal to the respective GPIO pin, which triggers the solenoid valve and pneumatic actuator to direct the object into the selected bin. After a delay of one second to allow proper actuation, the relay is turned off by setting the pin back to LOW, ensuring controlled and precise sorting operations.

B. Shape Detection Algorithm Using Contour Approximation

```python
# --------------- FUNCTION: SHAPE DETECTION ---------------
def detect_shape(contour):
    approx = cv2.approxPolyDP(contour, 0.04 * cv2.arcLength(contour, True), True)
    sides = len(approx)

    if sides == 3:
        return "Triangle"
    elif sides == 4:
        return "Rectangle"
    elif sides > 4:
        return "Circle"
    else:
        return "Unknown"
```

Fig. – 8 - Shape Detection Algorithm

This code defines a function detect_shape(contour) that identifies the geometric shape of an object based on its contour. The function uses OpenCV's contour approximation method, where cv2.approxPolyDP() simplifies the contour into a polygon with fewer vertices depending on the specified accuracy (0.04 times the contour perimeter calculated using cv2.arcLength()). The number of vertices (sides) of the approximated polygon is then determined using len(approx). Based on this value, the function classifies the shape: if the polygon has 3 sides, it is identified as a triangle; if it has 4 sides, it is considered a rectangle; and if it has more than 4 sides, it is assumed to be a circle. If none of these conditions match, the shape is labeled as "Unknown". This function is essential for object classification in the sorting system, as it enables decision-making based on the detected shape.

C. Main Processing Loop for Real-Time Image Acquisition and Object Detection

```
# --------------- MAIN LOOP ---------------
try:
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        # Convert to grayscale
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Blur to reduce noise
        blur = cv2.GaussianBlur(gray, (5, 5), 0)

        # Threshold
        _, thresh = cv2.threshold(blur, 100, 255, cv2.THRESH_BINARY)

        # Find contours
        contours, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

        for cnt in contours:
            area = cv2.contourArea(cnt)

            # Ignore small objects
            if area > 2000:
                shape = detect_shape(cnt)

               # Draw contour
               cv2.drawContours(frame, [cnt], -1, (0, 255, 0), 2)

               # Get bounding box
               x, y, w, h = cv2.boundingRect(cnt)

               # Display shape
               cv2.putText(frame, shape, (x, y),
                           cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 0), 2)
```

Fig. – 9 - Main Processing Loop for Real-Time Image Acquisition

This code represents the main processing loop of the system, where continuous image capture and object detection take place. The loop runs indefinitely using while True, and in each iteration, a frame is captured from the camera using cap.read(). If the frame is not successfully captured (ret is False), the loop terminates. The captured image is then converted into grayscale using cv2.cvtColor() to simplify processing by reducing color information. To minimize noise and improve detection accuracy, a Gaussian blur is applied using cv2.GaussianBlur(). The blurred image is then converted into a binary image through thresholding (cv2.threshold()), which separates objects from the background.

Next, contours are detected from the binary image using cv2.findContours(), where each contour represents a potential object. The program iterates through each contour and calculates its area using cv2.contourArea(). Small contours (noise) are ignored by applying a condition that only processes contours with an area greater than 2000. For valid contours, the shape is identified using the detect_shape() function. The detected contour is then drawn on the original frame using cv2.drawContours() for visualization. A bounding rectangle is created around the object using cv2.boundingRect(), and the identified shape name is displayed on the image using cv2.putText(). This loop continuously processes frames in real time, enabling detection and labeling of objects moving on the conveyor belt.

D. Decision-Making and Actuator Control Logic for Object Sorting

```
            # --------------- DECISION LOGIC ---------------
            # Example:
            # Rectangle → BIN 1
            # Circle → BIN 2

            if shape == "Rectangle":
                activate_bin(1)

            elif shape == "Circle":
                activate_bin(2)

            time.sleep(2)  # Delay to avoid multiple triggers

        cv2.imshow("Sorting System", frame)

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

Fig. – 10 -Decision-Making and Actuator Control Logic

This code segment implements the decision-making and output control part of the sorting system. Based on the detected shape of the object, the program decides which bin the object should be directed to. If the identified shape is a rectangle, the function activate_bin(1) is called, which triggers the actuator corresponding to Bin 1. Similarly, if the shape is detected as a circle, the function activate_bin(2) is executed to direct the object into Bin 2. After activating the respective bin, a delay of 2 seconds is introduced using time.sleep(2) to prevent multiple triggering for the same object as it remains in the camera frame.

Following the decision logic, the processed video frame with drawn contours and labels is displayed in a window titled "Sorting System" using cv2.imshow(), allowing real-time monitoring of the sorting process. The line cv2.waitKey(1) & 0xFF == ord('q') checks for a keyboard input, and if the user presses the 'q' key, the loop is terminated, effectively stopping the program execution.

XV. DESCRIPTION OF COMPONENTS

A. FULL ASSEMBLY

The system shown is a pneumatic conveyor-based automated material handling setup designed to transport and guide objects efficiently with minimal human intervention. It consists of a motor-driven conveyor belt that moves items forward, while pneumatic cylinders connected through air tubes are used to push or divert objects at specific points along the belt. The structure includes rollers, guide plates, and a rigid frame for smooth and stable operation, along with an electronic control unit that manages the timing and actuation of the pneumatic components. As objects travel on the conveyor, the system activates the actuators to direct them into desired paths, making it suitable for applications such as sorting, packaging, and assembly lines, ultimately improving productivity, accuracy, and automation in industrial processes.

Fig - 11 - Full Assembly

B. CAMERA

The camera module is one of the most important components in the Pneumatic Conveyor Sorting System integrated with Machine Vision. It acts as the primary sensing element responsible for capturing real-time images of objects moving on the conveyor belt. The accuracy and reliability of the entire sorting process depend significantly on the quality of image acquisition performed by the camera.

In this project, a Raspberry Pi Camera Module (8MP Sony IMX219 sensor) is used for capturing high-resolution images. This camera is specifically designed to interface directly with the Raspberry Pi through a CSI (Camera Serial Interface) ribbon cable. The module supports sill image capture as well as real-time video streaming, making it suitable for object detection and classification applications.

Fig – 12 - Camera Module

The Raspberry Pi Camera Module provides a resolution of up to 8 megapixels and supports video recording in HD formats. Its compact size, lightweight design, and easy connectivity make it ideal for embedded vision-based systems. The camera is mounted above the conveyor belt in a fixed position to ensure clear visibility of the objects passing through the inspection zone.

The main functions of the camera in this project include:

- Capturing continuous frames of objects moving on the conveyor.
- Providing real-time visual data to the Raspberry Pi.
- Ensuring proper focus and clarity for accurate feature extraction.
- Enabling color, shape, and size detection through image processing algorithms.

Proper positioning and alignment of the camera are essential for consistent image quality. The mounting height is selected in such a way that the entire object is visible within the frame without distortion. Adequate lighting conditions are also maintained to avoid shadows, glare, and noise in the captured images.

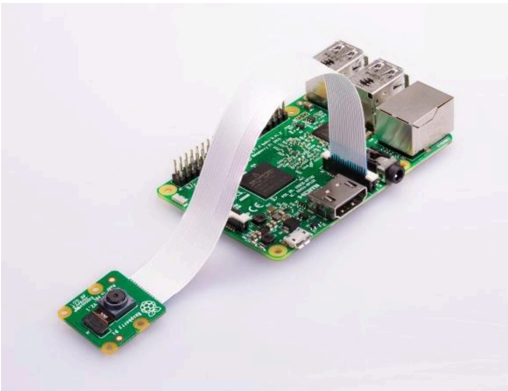

Fig – 13 - Camera Interface with Raspberry Pi

The camera module connects to the Raspberry Pi using a flat ribbon cable that transmits high-speed image data directly to the processor. Unlike USB cameras, the CSI interface provides faster data transfer and lower latency, which is important for real-time sorting operations.

- Sensor Type – Sony IMX219.
- Resolution – 8 Megapixels.
- Video Support – 1080p HD recording.
- Interface – CSI (Camera Serial Interface).
- Fixed Focus Lens..
- Compact and lightweight structure.

The use of this camera ensures reliable image acquisition with minimal delay. The captured images are processed using OpenCV-based algorithms to extract object features such as contours, colour intensity, and dimensional parameters. Based on this analysis, the system classifies the object and triggers the corresponding pneumatic actuator for sorting.

The camera module is selected due to its affordability, compatibility with Raspberry Pi, low power consumption, and ease of integration. It provides a cost-effective solution for developing industrial vision systems without requiring expensive industrial-grade cameras.

C. PROXIMITY SENSOR

The proximity sensor is an essential component in the Pneumatic Conveyor Sorting System integrated with Machine Vision. It is used to detect the presence of an object on the conveyor belt without physical contact. The sensor ensures proper synchronization between object movement, image capture, and actuation of the pneumatic system.

In this project, an inductive proximity sensor (cylindrical type) is used for detecting objects as they enter the inspection zone. Inductive sensors operate on the principle of electromagnetic induction and are capable of detecting metallic objects within a specified sensing range. When an object comes within the detection distance, the sensor generates an electrical output signal that is transmitted to the Raspberry Pi.

The proximity sensor plays a critical role in triggering the image capture process. Instead of continuously processing video frames, the system can use the sensor signal to initiate image acquisition only when an object is present. This improves processing efficiency and reduces unnecessary computational load.

The main functions of the proximity sensor in this project include:

- Detecting the presence of an object on the conveyor.
- Sending a trigger signal to the Raspberry Pi.
- Ensuring correct timing between object detection and image capture.
- Assisting in actuator timing for accurate sorting.

The sensor is mounted near the conveyor inspection zone at a suitable distance to ensure reliable detection. Proper alignment and stable mounting are necessary to avoid false triggering or missed detection. The sensing range is selected according to the object size and conveyor design.

Fig – 14 - Proximity Sensor Installation on Conveyor

Typical technical specifications of the inductive proximity sensor include:

- Operating Voltage – 6V to 36V DC.
- Output Type – NPN or PNP.
- Sensing Distance – 2 mm to 10 mm (depending on model).

- Non-contact detection mechanism.
- Compact cylindrical metal body.

The advantages of using a proximity sensor in this system are:

- Non-contact operation, reducing wear and tear.
- Fast response time.
- High reliability in industrial environments.
- Simple wiring and integration.

By incorporating the proximity sensor, the system achieves better coordination between mechanical movement and electronic processing. It enhances overall efficiency by ensuring that the machine vision module operates only when required, thereby improving real-time sorting performance.

Thus, the proximity sensor serves as a crucial detection element that supports accurate timing and smooth operation of the automated sorting system.

D. SOLENOID VALVE

The solenoid valve is a vital component in the Pneumatic Conveyor Sorting System integrated with Machine Vision. It functions as the control element that regulates the flow of compressed air to the pneumatic cylinders. Based on the classification signal received from the Raspberry Pi through the relay module, the solenoid valve directs air to actuate the corresponding cylinder for sorting action.

In this project, a 5/2 way pneumatic solenoid valve is used. This type of valve has five ports and two switching positions, allowing it to control the forward and return motion of a double-acting pneumatic cylinder. The valve operates electrically and converts electrical energy into mechanical movement to open or close internal air passages.

Fig – 15 - Pneumatic Solenoid Valve

The solenoid valve is connected between the air compressor and the pneumatic cylinder. Proper air pressure regulation is maintained to ensure smooth actuator motion without sudden jerks or instability.

Typical technical specifications of the solenoid valve include:

- Type – 5/2 Way Directional Control Valve.
- Operating Voltage – 12V or 24V DC.
- Working Pressure Range – 0.15 to 0.8 MPa.

- Response Time – Fast switching capability.
- Compact and durable metal body.

The advantages of using a solenoid valve in this project are:

- Fast switching speed.
- Reliable operation under continuous cycles.
- Easy integration with relay and microcontroller systems.
- Low maintenance requirements.

The solenoid valve ensures accurate synchronization between image classification and mechanical sorting action. Its quick response time and dependable operation make it suitable for repetitive industrial tasks.

Thus, the solenoid valve serves as the critical link between electronic control and pneumatic actuation, enabling efficient and automated sorting in the proposed system.

E. RELAY MODULE

The relay module is an important electrical interface component used in the Pneumatic Conveyor Sorting System integrated with Machine Vision. It acts as a switching device that allows the low-power output signal from the Raspberry Pi to control higher-power components such as solenoid valves and the conveyor motor. Since the Raspberry Pi operates at low voltage and limited current, direct connection to high-power devices is not safe. The relay module provides electrical isolation and safe control.

In this project, a 4-channel 5V relay module is used. Each relay acts as an electrically operated switch. When the Raspberry Pi sends a control signal, the corresponding relay coil is energized. This energization closes or opens the internal contacts, thereby allowing or interrupting the external power supply to devices like solenoid valves.

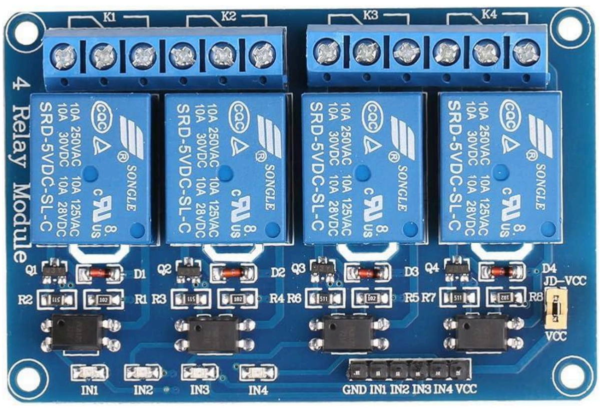

Fig – 16 - 4-Channel Relay Module

The relay module performs the following functions in the system:

- Acts as an interface between Raspberry Pi and high-power devices.
- Controls the activation of solenoid valves.
- Provides electrical isolation to protect the microcontroller.
- Ensures safe and reliable switching operation.

The relay module used typically includes optocouplers for additional isolation, which protects the Raspberry Pi from voltage spikes or electrical noise generated by inductive loads such as solenoid valves. This improves system safety and stability.

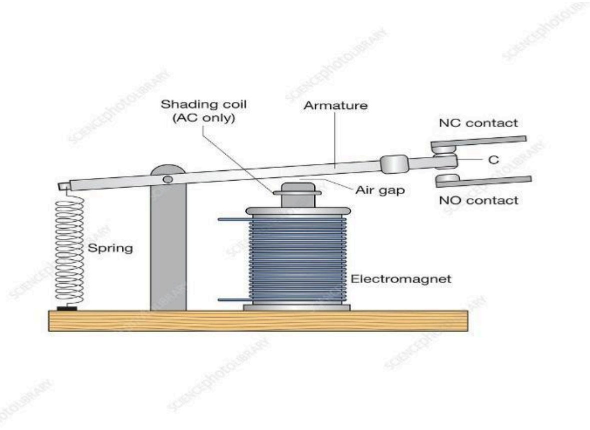

Fig – 17 - Relay Working Principle

Typical technical specifications of the relay module include:

- Operating Voltage – 5V DC.
- Number of Channels – 4.
- Trigger Type – Low-level or high-level trigger.
- Switching Capacity – Up to 10A at 250V AC or 30V DC.
- Optocoupler Isolation for protection.

The advantages of using a relay module in this project are:

- Safe control of high-power components.
- Electrical isolation between control and power circuits.
- Simple interfacing with Raspberry Pi GPIO pins.
- Reliable operation for repeated switching cycles.

The relay module plays a crucial role in ensuring that electronic control signals from the image processing unit are effectively converted into physical actuation commands. It forms the electrical bridge between the digital processing system and the pneumatic hardware .Thus, the relay module ensures safe, controlled, and synchronized operation of the sorting mechanism within the automated system.

F. RASPBERRY PI

The Raspberry Pi serves as the central processing and control unit of the Pneumatic Conveyor Sorting System integrated with Machine Vision. It performs image processing, decision-making, and system control operations. The Raspberry Pi receives image data from the camera module, processes it using programmed algorithms, and generates appropriate output signals to control the relay modules and pneumatic actuators.

In this project, a Raspberry Pi 4 Model B (or equivalent model) is used due to its sufficient processing capability, compact size, low power consumption, and compatibility with camera and GPIO interfaces. It operates as a small embedded computer capable of running a full operating system and executing image processing programs developed in Python using OpenCV.

Fig – 18 - Raspberry Pi Board

The Raspberry Pi performs the following major functions in the system:

- Captures image data from the camera module.
- Processes images using computer vision algorithms.
- Sends control signals to relay modules.
- Coordinates timing between detection and actuation.

Typical technical specifications of Raspberry Pi 4 Model B include:

- Processor – Quad-core ARM Cortex.
- RAM – 2GB / 4GB / 8GB (depending on model).
- GPIO Pins – 40-pin header.
- CSI Port – Dedicated camera interface.
- USB and HDMI support.
- Operating Voltage – 5V DC.

The advantages of using Raspberry Pi in this project are:

- Compact and lightweight design.
- Affordable and widely available.
- Supports high-level programming languages like Python.
- Easy integration with camera and GPIO-based hardware.
- Capable of real-time image processing.

The Raspberry Pi enables intelligent automation by combining sensing, processing, and control within a single platform. It eliminates the need for expensive industrial controllers while maintaining reliable system performance .Thus, the Raspberry Pi acts as the brain of the proposed sorting system, coordinating image analysis and pneumatic actuation to ensure accurate and efficient automated sorting.

G. DOUBLE ACTING PNEUMATIC CYLINDER

The double acting pneumatic cylinder is a critical actuation component in the Pneumatic Conveyor Sorting System integrated with Machine Vision. Unlike single acting cylinders that use compressed air only for one direction of motion and a spring for return, the double acting cylinder uses compressed air on both sides of the piston for both extension and retraction strokes. This provides greater control, consistent force in both directions, faster response, and more reliable and repeatable performance in high-cycle sorting applications.

In this project, double acting cylinders are mounted at the sorting positions along the conveyor belt. When the Raspberry Pi classifies an object and sends a signal through the relay to the solenoid valve, compressed air is admitted to the forward port of the cylinder, causing the piston rod to extend and physically push the object off the conveyor into the designated sorting bin. After the sorting action is complete, the solenoid valve switches position, admitting compressed air to the return port and retracting the piston rod to its original position. This bidirectional air supply ensures swift and powerful actuation in both directions without relying on a spring return mechanism.

Fig. – 19 - Double Acting Pneumatic Cylinder

The cylinder body is constructed from anodized aluminum for corrosion resistance and light weight. The piston rod is made from chrome-plated steel to ensure smooth, low-friction movement and long service life. Rubber seals on the piston prevent internal air leakage between the two chambers, maintaining pressure integrity and ensuring efficient conversion of pneumatic energy into linear mechanical force.

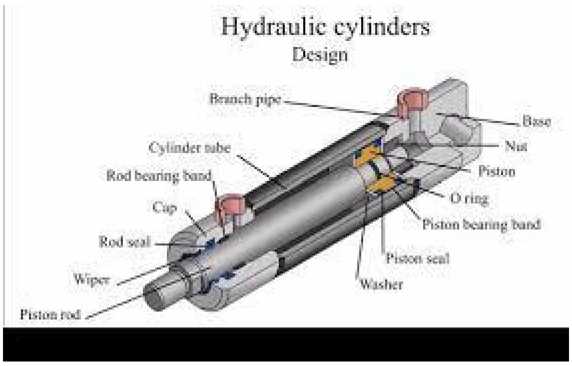

Fig. – 20 - Double Acting Cylinder Cross Section

Typical technical specifications of the double acting pneumatic cylinder used in this project include:

- Type: Double Acting, Spring-less.
- Cylinder Bore Diameter: 25 mm.
- Piston Rod Diameter: 10 mm.
- Stroke Length: 50 to 100 mm.
- Operating Pressure: 1 to 10 bar (working range 4 to 6 bar).
- Cylinder Body Material: Anodized aluminum.
- Piston Rod Material: Chrome-plated stainless steel.
- Port Size: G1/8 inch.
- Sealing Material: NBR (Nitrite Butadiene Rubber).
- Cushioning: Adjustable air cushioning at both ends.

H. SERVO MOTOR

The servo motor is an important electromechanical component integrated into the Pneumatic Conveyor Sorting System to provide precise angular positioning for the sorting gate mechanism. While the double acting pneumatic cylinder performs the linear sorting action, the servo motor controls the angular orientation of deflector gates or bin selector flaps, directing objects accurately into one of multiple designated sorting bins based on the classification result from the Raspberry Pi.

Fig. – 21 - Servo Motor

In this project, a standard hobby-grade servo motor (such as the SG90 or MG995 type) is used, interfaced directly with the GPIO pins of the Raspberry Pi. The Raspberry Pi generates PWM signals with pulse widths varying between 0.5 ms and 2.5 ms, corresponding to angular positions typically ranging from 0 to 180 degrees. The servo motor is mounted at the sorting gate position on the conveyor frame. When the Raspberry Pi classifies an object and determines the target bin, it generates the corresponding PWM signal, causing the servo to rotate to the precise angle that opens the gate toward the correct bin. After the object passes, the servo returns to its neutral position, ready for the next sorting cycle.

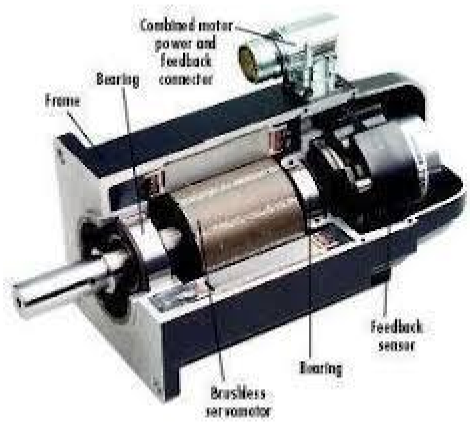

Fig. - 22 - Servo Motor Internal Structure

The use of a servo motor provides significant advantages over conventional fixed mechanical deflectors:

- Programmable angular positioning allows dynamic bin selection without mechanical modification.
- Multiple sorting categories can be accommodated by defining different angle positions in software.
- High repeatability ensures consistent gate positioning across thousands of sorting cycles.
- Fast response time (typically under 200 ms for 60 degrees of rotation) is compatible with conveyor belt speeds used in the system.
- Direct interfacing with the Raspberry Pi GPIO through PWM eliminates the need for additional motor drivers for low-torque applications.

Typical technical specifications of the servo motor used in this project include:

- Type: Position-controlled servo motor (PWM-driven).
- Operating Voltage: 4.8V to 6V DC.
- Stall Torque: 1.8 kgf·cm (at 4.8V) to 2.5 kgf·cm (at 6V).
- Operating Speed: 0.1 s / 60 degrees (at 4.8V).
- Angular Range: 0 to 180 degrees.
- PWM Control Signal: 20 ms period; pulse width 0.5 ms to 2.5 ms.
- Gear Type: Plastic gears (SG90) or metal gears (MG995) for heavy-duty applications.
- Connector: Standard 3-wire (VCC, GND, Signal).

The servo motor is powered by the 5V supply from the Raspberry Pi power rail or an external 5V regulated supply when higher torque is required. To avoid voltage drops caused by servo load affecting the Raspberry Pi, it is recommended to use a dedicated 5V

XVI. RESULTS AND DISCUSSION

The developed Pneumatic Conveyor Sorting System was experimentally tested under controlled laboratory conditions to evaluate its performance in terms of sorting accuracy, response time, and operational stability. Multiple test trials were conducted using objects of varying sizes and shapes. The system achieved an average sorting accuracy of approximately 95%, demonstrating reliable object classification using machine vision techniques. The integration of OpenCV-based image processing with Raspberry Pi enabled real-time detection with minimal processing delay.

The pneumatic actuation system showed rapid response characteristics, with efficient cylinder movement upon receiving control signals from the controller. The use of solenoid valves and regulated air pressure (5–7 bar) ensured consistent and precise object diversion. During continuous operation testing, the system maintained stable performance without significant mechanical or processing errors.

Compared to conventional manual sorting methods, the proposed system significantly reduced human intervention, minimized sorting errors, and improved throughput. The modular and compact design makes it suitable for small and medium-scale industrial applications. Overall, the results confirm that the integration of machine vision with pneumatic control provides an efficient and cost-effective automated sorting solution.

XVII. CONCLUSION

The proposed Pneumatic Conveyor Sorting System integrated with Machine Vision successfully demonstrates an efficient and cost-effective solution for automated object classification and sorting. The system combines real-time image processing using OpenCV with high-speed pneumatic actuation to achieve accurate object detection based on size and shape. Experimental results indicate an average sorting accuracy of approximately 95%, confirming reliable system performance under continuous operation.

The integration of Raspberry Pi-based control with pneumatic cylinders ensures rapid mechanical response and reduced manual intervention. The compact and modular design enhances scalability and suitability for small and medium-scale industries such as packaging, food processing, and recycling.
Future enhancements may include deep learning-based classification for improved accuracy, IoT-enabled monitoring for remote supervision, and large-scale industrial deployment for increased throughput and operational efficiency.

References

[1] S. Kumar and R. Singh, (2022) , “Automated Conveyor Sorting Using Machine Vision Techniques,” IEEE Transactions on Industrial Electronics, vol. 69, no. 4, pp. 3456–3464, 2022.

[2] M. Patel and S. Desai,(2022.), “Performance Evaluation of Conveyor-Based Sorting Systems,” International Journal of Mechanical Engineering and Robotics Research, vol. 12, no. 5, pp. 210–218, 2022.

[3] D. Johnson (2023), “Application of OpenCV in Industrial Automation,” Journal of Automation and Control Engineering, vol. 10, no. 1, pp. 15–22, 2023.

[4] R. Rajesh and T. Arun, (2023)“Real-Time Image Processing for Industrial Object Classification,” International Journal of Advanced Manufacturing Technology, vol. 118, no. 3, pp. 1125–1136, 2023.

[5] A. Verma, P. Sharma, and M. Gupta,(2024) “Low-Cost Vision-Based Object Sorting System Using Raspberry Pi,” IEEE Access, vol. 11, pp. 55678–55689, 2024.

[6] J. Lee and K. Park,(2025), “Design and Analysis of Pneumatic Actuation Systems for Industrial Automation,” IEEE Industry Applications Magazine, vol. 30, no. 2, pp. 44–52, 2025.