International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

AI-Driven Sericulture Automation for Optimal Silkworm Rearing

Dr M Shilpa[1], Samarth[2], Ullas Raj B. R[3], Suman B.S[4], Haradik[5]

[1,2,3,4,5]Department of Information Science and Engineering, Bangalore Institute of Technology, Bangalore, India

ABSTRACT:
This paper is an AI-powered sericulture automation system for optimal silkworm rearing. Combining synthetic intelligence and IoT to revolutionize sericulture With computerized temperature control capabilities for robust farming conditions. Precise feeding mechanism to ensure consistent nutrition Integrated anomaly detection engine based on deep expertise to quickly recognize infections State-of-the-art machinery for producing top-notch silk by increasing productivity. Reduce human intervention and reduce losses Ensure environmentally friendly, sustainable and scalable practices.

1. INTRODUCTION

Sericulture has been a cornerstone of the textile enterprise for hundreds of years. However, traditional sericulture techniques are hard work in depth. It is useless and quite dependent on environmental and biological conditions. Temperature changes Irregular feeding schedules And disorder outbreaks may have a huge impact at the fitness of silkworms. Reduce productivity and fine?

In a modern technological era, integrating modern innovations like Artificial Intelligence (AI) and the Internet of Things (IoT) into sericulture offers transformative potential. These technologies provide precision, real-time monitoring, and adaptive decision-making, which can significantly improve the efficiency and reliability of silkworm rearing. By using AI and IoT, this project aims to address the key challenges faced by traditional sericulture practices and redefine the industry's operational standards.

This project is an AI-powered sericulture automation system for optimal sericulture. Addressing the challenges faced by traditional sericulture by developing intelligent automation solutions.

The system focuses on three key points:

1. Automatic temperature control:

Maintains appropriate environmental conditions such as temperature and humidity. It is important for the health and productivity of silkworms. The system uses IoT-enabled sensors to monitor and control these conditions.

2. Automatic Feeding Mechanism:

Feeding is essential for consistent silkworm growth. The proposed solution uses an automated catering system to ensure timely and accurate food delivery. This reduces manual labor and errors.

3. Disease Detection System:

Silkworm diseases can spread quickly and cause great damage. Deep learning models use a CNN algorithm to detect early signs of disease. This allows timely intervention and risk reduction.

The integration of these components creates an approach to optimizing silkworm rearing. The system not only enhances productivity and improves silk quality but also reduces dependency on human labor,

making sericulture more sustainable and scalable. Additionally, the implementation of real-time monitoring and AI-driven automation minimizes risks associated with environmental variations and disease outbreaks, fostering greater resilience in the sericulture industry.

This project represents a significant step toward modernizing sericulture practices, empowering farmers with advanced tools to achieve results and meet the growing global demand for silk. By mixing traditional agricultural knowledge with cutting-edge technology, creates the way for a new era of precision sericulture, ensuring economic viability and long-term sustainability.

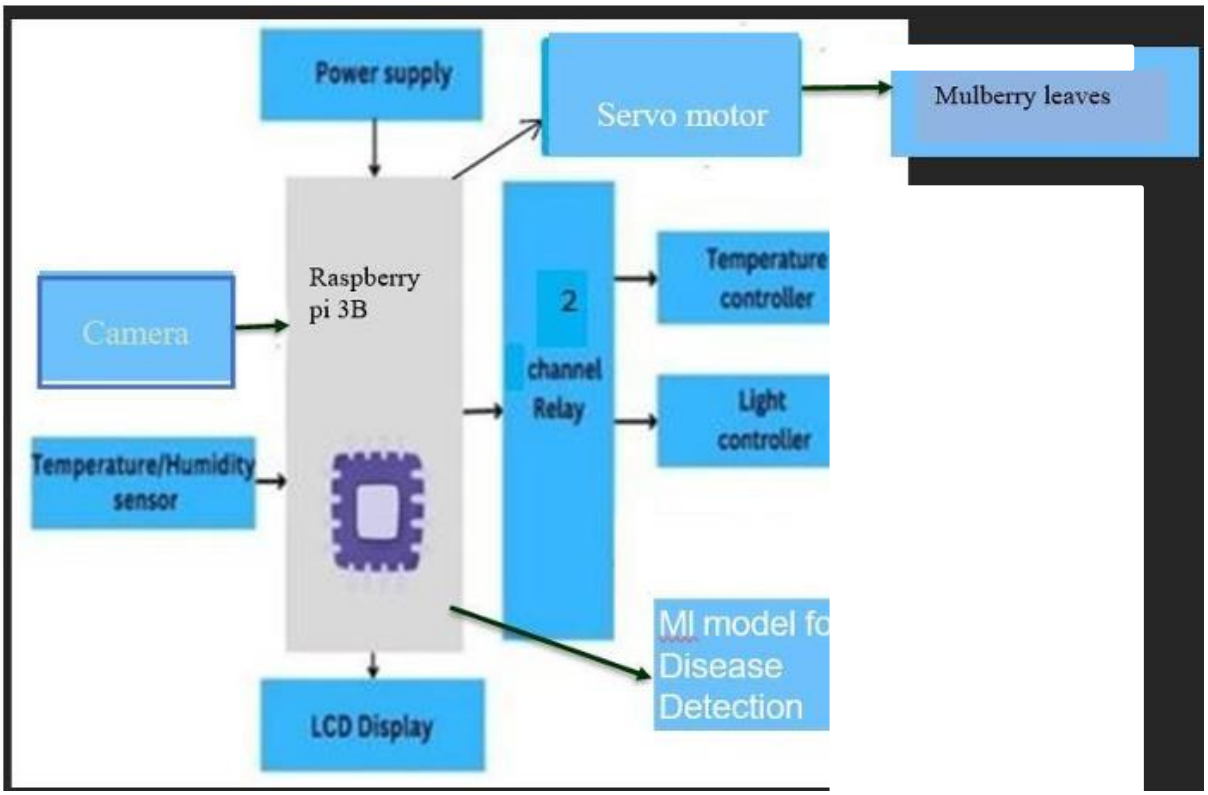


Figure 1. System architecture Diagram

3. METHOD

In this section, we provide a detailed description of the methodology followed during the course of the implementation of the proposed approach methodology for AI-Driven Sericulture Automation for Optimal Silkworm Rearing, with explanations of each module.

1. Environmental Control Using DHT22 and Raspberry Pi 4B

To maintain optimal temperature and humidity conditions for silkworm rearing

Components Used:

- DHT22 Sensor
- Raspberry Pi

The DHT22 sensor continuously monitors temperature and humidity in the rearing chamber and sends the data to a Raspberry Pi 4B. A Python script processes the data in real-time. If the temperature goes outside the ideal range (23–28°C), the system activates light. The system maintains environmental conditions by using an automated feedback loop with pre-defined thresholds for temperature and humidity.

2. Automated Feeding Mechanism Using Servo Motor

To ensure consistent feeding of silkworms, reducing manual intervention and waste.

Components Used:

- Feeding Box
- Raspberry Pi
- Servo Motor

The feeding box is filled with mulberry leaves . The Raspberry Pi is programmed with a feeding schedule based on the silkworms' growth stages. At specific intervals, the Raspberry Pi signals the servo motor to open the feeding box and release a leaves for feeding . After dispensing, the motor closes the box to prevent contamination . A sensor monitors the feeding process to ensure everything functions smoothly.

3. Disease Detection Using CNN Algorithm:

To identify early signs of diseases in silkworms using image recognition technology

Components Used:

- Camera
- Raspberry Pi
- A camera captures images of the silkworms in their rearing environment. These images are sent to the Raspberry Pi, where they are pre-processed for analysis. A pre-trained CNN model is used to detect early signs of diseases like fungal infections, discoloration, or abnormal behavior.
- For training the model, a dataset of healthy and diseased silkworm images is collected. Data augmentation techniques are applied to enhance the dataset. The CNN model is trained using frameworks like TensorFlow, with layers optimized for feature extraction and classification.

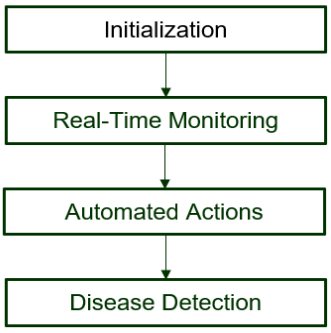


Fig 2. System Workflow

- Once an image is analyzed, the system classifies the silkworms as healthy or diseased.

Initialization:

- The Raspberry Pi initializes all connected components (DHT22 sensor, servo motor, and camera).
- The environmental monitoring system and feeding mechanism are set to operate autonomously.

Real-Time Monitoring:

- Environmental data from the DHT22 sensor is continuously monitored and regulated.
- Periodic images are captured by the camera and analyzed for diseases using the CNN model.

Automated Actions:

- Environmental controls adjust temperature and humidity as needed.

- The feeding system dispenses food at predefined intervals.

Disease Detection:

- Camera captures images of silkworms, which are then pre-processed for analysis.
- The CNN model detects diseases.

4. IMPLEMENTATION

The following sections describe the procedural implementation. The implementation of the AI-Driven Sericulture Automation for Optimal Silkworm Rearing involves setting up hardware, software, and integrating various components to automate the silkworm rearing process.

The implementation follows these key steps:

Hardware Setup:

- **DHT22 Sensor:** Monitors temperature and humidity for optimal silkworm rearing.
- **Raspberry Pi 4B:** Central controller for processing data and managing systems.
- **Servo Motor:** Controls the automated feeding system.
- **Camera:** Captures images for disease detection.

Software and Integration:

- **Temperature and Humidity Control:** Python scripts regulate the environment based on sensor data.
- **Automated Feeding:** Raspberry Pi schedules and controls the servo motor to dispense feed.
- **CNN-Based Disease Detection:** Captures images, analyzes them using a trained CNN, and diseases are detected.

Testing and Calibration:

- Sensor, feeding mechanism, and CNN model are determined and tested for accuracy.

Deployment and Monitoring:

- The system is deployed in a sericulture environment and continuously monitored for performance.

3.1 Code:

```
import Adafruit DHT
import smbus2
import time
import RPi. GPIO as GPIO
import numpy as np
import cv2
import tflite_runtime.interpreter as tflite
# Configuration for DHT Sensor, Fan, Servo, and LCD
SENSOR = Adafruit DHT.DHT22
DHT PIN = 4
I2C ADDR = 0x27
LCD WIDTH = 16
LCD CHR = 1
LCD CMD = 0
LCD LINE 1 = 0x80
LCD LINE 2 = 0xCO
ENABLE = 0b00000100
```

```
FAN PIN = 17
SERVO PIN = 18
FAN PINI = 16
GPIO. setmode (GPIO.BCM)
GPIO. setup(FAN_PIN, GPIO.OUT)
GPTO setuin ( FAN PTN1 _ GPTO QUIT)
GPIO . output ( FAN_PIN, GPU. LOW)
pwm_servo = GPIO. PWM(SERVO_PIN, 50)
pwm_servo.start(0)
bus = smbus2 . SMBus (1)
# Initialize TFLite Interpreter
class_names = ["Bacterial Diseases", "Fungal Diseases", "Healthy", "Protozoan”]
try :
interpreter = tflite. Interpreter(model_path="modell. tflite")
interpreter. allocate_tensors ()
input_details = interpreter.get_input_details ()
output_details = interpreter.get_output_details()
        input_data = preprocess_image (frame)
        interpreter. set_tensor(input_details [0] ['index' ], input_data)
        interpreter. invoke ()
        predictions = interpreter. get_tensor(output_details [0] ['index' ] )
        predicted_class = np. argmax (predictions)
        confidence = np.max(predictions) * 100
        prediction_ccc, preddction = predict_by_color(frame)
        predicted_label = class_names [prediction_ccc]
        based_label = f"Prediction: {predicted_label} ({confidence})"
        predicted_label = class_names [predicted_class ]
        cv2.putText (frame, based_label, (10, 30), cv2. FONT_HERSHEY_SIMPLEX, 1, (0,
        resized_frame = cv2. resize(frame, (600, 600))
        cv2.imshow("Silkworm Disease Detection", resized_frame)
        if cv2.waitKey (1) & 0xFF == ord('q' ) :
        break
        i+=1
        print (i)
        if i == 10:
        break
        cap. release()
        cv2. destroyAllWindows ( )
        def handle_dht_servo_lcd() :
        lcd init()
        countdown = 50
        while True:
        humidity, temperature = Adafruit_DHT. read_retry (SENSOR, DHT_PIN)
        if humidity is not None and temperature is not None:
        temp_str = f"T : {temperature : . 1f}C H: {humidity : . 1f}%"
        lcd_string(temp_str, LCD_LINE_1)
        minutes, seconds = divmod (countdown, 60)
        timer str = f"LEAF IN : {minutes : 02} : {seconds : 02}"
        lcd_string(timer_str, LCD_LINE_2)
        countdown -= 1

camera_available = True
```

```
except :
print ("TFLite Model failed to load. Proceeding with DHT sensor tasks.")
camera_available = False
def lcd_byte(bits, mode) :
high_bits = mode | (bits & 0xF0) | 0x08
low bits = mode | ((bits << 4) & 0xF0) | 0x08
bus.write_byte(I2C_ADDR, high_bits)
lcd_toggle_enable(high_bits)
bus.write_byte(I2C_ADDR, low_bits)
lcd_toggle_enable(low_bits)
det Lcd_toggle_enable ( bits) :
time.sleep(0.0005)
bus.write_byte(I2C_ADDR, (bits & ~ENABLE) )
time.sleep(0.0005)
bus.write_byte(I2C_ADDR, (bits | ENABLE) )
time.sleep(0.0005)
def lcd_init():
lcd_byte(0x33, LCD_CMD)
lcd_byte(0x32, LCD_CMD)
lcd_byte(0x06, LCD_CMD)
lcd_byte (0x0℃, LCD_CMD)
lcd_byte(0x28, LCD_CMD)
lcd_byte(0x01, LCD_CMD)
time.sleep(0.0005)
def lcd_string(message, line) :
message = message. ljust (LCD_WIDTH, " ")
lcd_byte(line, LCD_CMD)
for i in range (LCD_WIDTH) :
lcd_byte(ord (message[i] ), LCD_CHR)
def preprocess_image (img) :
img = cv2. resize(img, (150, 150) )
img = np. expand dims (img, axis=0) / 255.0
def predict_by_color( frame) :
# Convert frame to HSV for better color detection
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
# Define color ranges (in HSV)
color_ranges = {
'bact': ((0, 0, 200), (180, 25, 255)),
'vir': ((0, 0, 0), (180, 255, 50)),
healt': ((36, 25, 25), (70, 255, 255)),
'fung': ((0, 50, 50), (10, 255, 255)),
'lsl': ((90, 50, 70), (128, 255, 255) )
# Calculate pixel count for each color
```

```
color_counts = {color: cv2. inRange(hsv, np. array(lower), np. array(upper) ) . sum( )
dominant_color = max(color_counts, key=color_counts. get)
# Map color to prediction
color_to_prediction = {
'bact': 3,
'vir': 2,
'healt': 3,
'fung': 4,
'lsl': 0
return color_to_prediction[dominant_color], dominant_color
def silkworm_disease_detection () :
i=0
cap = cv2. VideoCapture (0)
while cap. isOpened() :
ret, frame = cap.read ()
if not ret:
break
if countdown == 0:
pwm_servo. ChangeDutyCycle (12)
# 7% duty cycle for 90 degrees
time. sleep(5) # Wait for 5 seconds before restarting timer
pwm_servo. ChangeDutyCycle(2) # Stop the servo
time.sleep(1) # Wait for 5 seconds before restarting timer
pwm_servo. ChangeDutyCycle(0) # Stop the servo
countdown = 30
break
GPIO. output (FAN_PIN, GPIO.HIGH if temperature > 31 else GPIO.LOW)
GPIO. output (FAN_PIN1, GPIO.HIGH if temperature < 30 else GPIO.LOW)
else:
lcd_string ("Sensor Error", LCD_LINE_1)
time.sleep (1)
break
GPIO.output (FAN_PIN, GPIO.HIGH if temperature > 31 else GPIO. LOW)
GPIO. output (FAN_PIN1, GPIO.HIGH if temperature < 30 else GPIO. LOW)
else:
lcd_string ( "Sensor Error", LCD_LINE_1)
time.sleep(1)
#try:
while 1:
if camera_available:
silkworm_disease_detection()
handle_dht_servo_lcd()
if 1:
```

```
lcd_string ( "Goodbye! ", LCD_LINE_1)
GPIO.output (FAN_PIN, GPIO.LOW)
GPIO. output (FAN_PIN1, GPIO.LOW)
pwm_servo.stop()
GPIO.cleanup()
cv2. destroyAllWindows ( )
```

RESULTS

4.1 Disease Detection Result

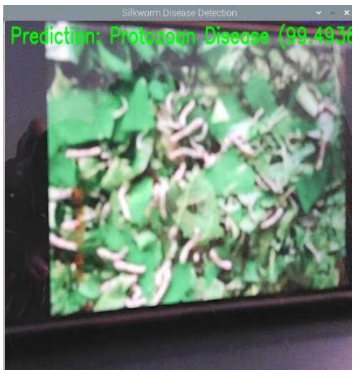


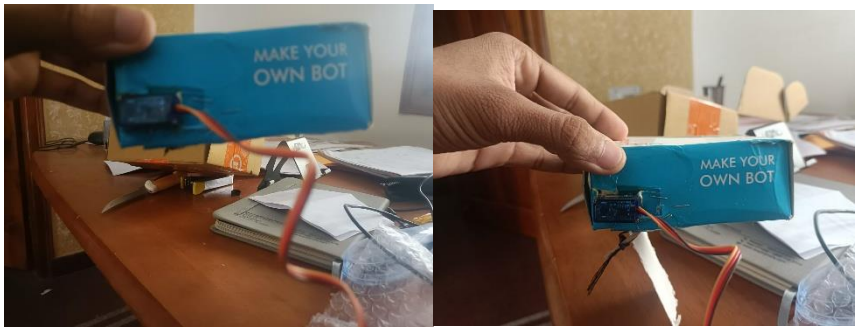
Fig 4.1.1 Protozon Disease

Fig 4.1.2 Viral Disease

4.2 Temperature Control Result:

Fig4.2.1 temperature control

4.3 Silkworm Feeding Result:



Closed box Open box

Figure 4.3.1 Silkworm Feeding

5. CONCLUSION

In this paper, we presented an approach to enhance traditional sericulture through AI and automation. By integrating temperature and humidity control, automated feeding, and CNN-based disease detection, we optimized silkworm rearing conditions, improving health and silk yield. The system reduces human Effort, ensures real-time monitoring, and enables early disease detection, offering an efficient solution for modernizing sericulture and increasing productivity.

6. REFERENCES

1. 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON) Pilot Implementation of Efficient Automation in Sericulture Farms Using Internet of Things (IoT).
2. IEEE 802.15.4-2020, Wireless Personal Area Networks (WPANs), IEEE, 2020.
3. J. Smith, “Introduction to IoT in Agriculture,” Tech Innovations. [Online]. Available: http://www.techinnovations.com/iot-agriculture. [Accessed: Aug. 1, 2024].
4. IoT-based Smart Sericulture System" (2018) – Discusses environmental monitoring of silkworms using IoT sensors.
5. IoT-Enabled Precision Sericulture: A Review" (2020) – A review of IoT in sericulture for optimization and data-driven farming.
6. "Smart Agriculture: IoT Applications" (2021) – Discusses IoT applications in various agricultural sectors, including sericulture.
7. McKinsey report on IoT's impact on agriculture (2021), with mention of sericulture.
8. International Conference on Smart Agriculture and IoT (ICSAI) – Focuses on IoT in farming, including sericulture.