Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

ISSN NO. 2320-5407

Article DOI: 10.21474/IJAR01/22204
DOI URL: http://dx.doi.org/10.21474/IJAR01/22204

RESEARCH ARTICLE

INTELLIGENT EXPIRY DATE RECOGNITION AND ALERT SYSTEM

Kommalapati Abhinaya Sri[1], Thottempudi Sree Varsha[1], Thikkavarapu Chandra Sekhar Reddy[1], Katragadda Mounika[1], Dasari Glory[1], Mukku Niranjan[1] and Dr Gnana Sagaya Raj[2]

1. UG Student, CSE-AI & DS, QIS College of Engineering and Technology (A), Ongole, Andhra Pradesh, India.
2. Associate Professor, Department of Mechanical Engineering, QIS College of Engineering and Technology(A), Ongole, Andhra Pradesh, India.

Manuscript Info

Manuscript History
Received: 04 September 2025
Final Accepted: 06 October 2025
Published: November 2025

Key words:-
Expiry date recognition, QR code, Arduino, LED alert, OCR, IoT, retail automation.

Abstract

In supermarkets such as D-Mart, Reliance Smart, and More, customers often purchase packaged items, which may have QR codes providing product information but customers tend to miss out on the expiry dates. The consumption of expired products can cause health problems and dissatisfaction to the customer. In this study, a smart expiry date recognition and alert system is proposed which will read the expiry date either from the product QR code or printed text using OCR and provide a visual alert via an LED [6]. The system consists of a microcontroller, such as Arduino a QR code scanner, and a real-time clock module OCR to determine the validity of the product. A red LED indicates the product is expired, an orange LED indicates the product has less than one month of remaining shelf life, and a green LED indicates the product is safe to use. This system can reduce the manual effort of expiry date checking, limit the usage of expired products, and improve consumer safety and trust with retail environments.

Introduction:-

In the current rapid retail environment, consumers often purchase packaged food or pharmaceutical products from grocery stores and supermarkets without adequate review of the expiration date[1]. The result may be consumption of expired products which could pose health issues, unenforceable customer concerns, and financial loss for the retailer [1],[5]. Additionally, timely awareness of expired items on the shelf may be difficult because reviewing dates on thousands of items takes staff time and is subject to human error [2],[3]. Retailers and warehouse workers face difficulties in efficiently reviewing expired or near-expired food items, ultimately creating product waste and regulatory violations. In a time of increased automation and the Internet of Things (IoT) equipment in smart retail environments, it is necessary to have a low-cost automated reliable method to evaluate product expired product items and alert users in real time [3]. Current systems typically utilize QR codes for billing and product identification purposes; however, they do not provide a validation method for expired products and do not produce any alerts. To address this limitation, the proposed Intelligent Expiry Date Recognition and Alert System offers an automated means of reading expiry information for products from its packaging using QR codes or printed text from a product via Optical Character Recognition (OCR) [17]. The system utilizes a microcontroller, such as an Arduino or Raspberry Pi, a QR-code scanner, and a Real-Time Clock (RTC) module, to compare the current date with the expiry date extracted from product packaging, to provide the user with a notification of the product's remaining shelf life, based on an LED indicator:

- - Red for expired,
- - Orange for near expiry,
- - Green for safe to consume.

The system can be implemented at billing counters, storage areas, or a retail shelf to help both consumers and retail staff make informed decisions regarding product safety. The proposed solution aids in automating the system without requiring human intervention, enhances food and medicine safety, and reduces wastage of food or medicine products. The proposed solution also allows for scale, providing an opportunity for the system to later integrate into IoT, smart retail systems, for example, mobile alerts, or cloud-based tracking of expiry dates.

Literature Survey:-

Research on expiry monitoring, as well as related food- safety systems, covers systems based on QR/barcode technologies, systems that use OCR approaches for date recognition, Internet of Things or edge-monitoring systems, as well as data-driven (i.e., machine learning) systems. Relevant trends and representative works in the literature are summarized below.

QR-code and barcode systems:-

QR, and barcode technologies offer a robust mechanism for capturing product metadata and have been used in retail broadly for product identification and checkout [22]. Tiwari provides an introductory review characterizing the QR-code technology, including the advantages of QR codes—high capacity, error correction, and ability to scan in all directions—along with dependencies on appropriate scanners and other issues of security and privacy. Some applied systems examine QR/barcode scanning as it relates to inventory and/or billing systems with supporting cloud alerts and basic expiry monitoring. In general, the dependence on QR codes and/or a connected network for these systems creates a practical barrier for implementation in diverse retail environments.

OCR and image-based expiry recognition:-

OCR techniques aim to extrapolate printed manufacturing (MFD) and expiry (EXP) dates from surfaces of packages where QR codes are missing or incomplete[10]. The "Recognition of Expiration Dates written on Food Packages" employs Tesseract OCR with image preprocessing and classification to create an mobile or embedded product. A more complex pipeline may include object detect (e.g., FCOS) with sequence recognizers (CRNN with CTC loss) for text localization and recognition on product surfaces; following this approach could achieve very high accuracy (in studies, this may be reported as high as ~97% when studies have been optimized). Limitations of OCR techniques generally include poor performance with slanted, reflective, low-contrast, dot- matrix, or thin-font printing, as well as compromised recognition due to lighting conditions or camera motion—all of which are practical issues within retail settings. Additionally, OCR accuracy can be further affected by variations in packaging design, text placement, and inconsistent expiry date formats across different brands and products.

ML, analytics and cloud solutions:-

Several studies have proposed cloud-technology driven architectures embedding expiry data towards analytics, alerts and inventory decisions. Machine learning (Random Forest, SVMs, CNNs) has been utilized for invoice/text mining to detect food-safety issues and visual classification tasks involving expiry recognition[13]. Cloud-enablement may provide for centralized monitoring, analytics and notification services; however, it may rely on the availability of a network connection and has latency issues and there's practical challenges with data privacy, scalability-this is compounded by the need for datasets with labels prior to uptake[11].

Research studies of usability and behavior:-

Surveys assess consumer awareness of and behavioral reactions toward expiration information. Again, as can be seen in social science work, technical solutions must also consider social factors (pattern of visibility of alerts, methods of integration into check-out processes, adoption by employees) to be effective. Evaluating only the technical accuracy of your technologies will not provide value without understanding system integration and user workflows[14].

Summary of gaps and implications for this work:-

The body of literature reviewed presents distinct progress across several areas—high-accuracy OCR models, practical QR-based systems, IoT edge solutions, and cloud analytics— but also shows repeated limitations: [17] OCR fragility on poor prints/lighting, dependence on availability of QR codes, reliance on cloud/network, and limited capabilities for real- time alerting at the point of sale or shelf. Systems currently in existence either emphasize (a) high-compute ML solutions that cannot be deployed easily on low-cost edge hardware, or (b) simple QR/cloud solutions that do not work in the absence of QR codes or when QR codes are poorly printed.

Project positioning:

The suggested Intelligent Expiry Date Recognition and Alert System is designed to improve any existing reliability and usability inadequacies in expiration date detection by using QR code decoding (if available) and OCR-based recognition in case the QR code is not applicable. It is developed for use on low-cost microcontrollers such as Arduino or Raspberry Pi, using low-level decision logic to optimize for low-latency expiry detection in real time without the requirement for advanced computing capabilities. In the retail checkout or on the storage shelf, it provides immediate, visual LED alerts for expired or soon-to-expire items, all while significantly reducing the manual effort associated with expiration date tracking. The hybrid approach of using computer vision techniques and embedded IoT hardware establishes a practical balance among accuracy, responsiveness, and cost, in a scalable way, to be applied to various retail or storage environments such as groceries. In the end, the system will deliver reliable alerting, intelligence, and efficiency in expiry management by increasing food safety, minimizing waste, and simplifying inventory observation.

Methodology:-

System presented here, called "Intelligent Expiry Date Recognition and Alert System," combines an Arduino microcontroller and a Python-based image processing module to automatically detect a product's expiry date and alert the user with visual signals via LED lighting. Real-time expiry detection is enabled by the combination of concepts in computer vision systems as well as embedded control systems.

System Architecture:

The architecture of the system has two main subsystems or units:

1. Image Processing and Recognition Unit (Python + USB webcam + OpenCV)[23]
2. Embedded Control and Alert Unit (Arduino + LEDs)

A conventional USB web camera is used to take product photos displaying expiration date or manufacturing date details. The camera is the primary input device, continually sending frames to the system. Each frame is pre-processed via OpenCV functions to improve image quality and text detection. The pre-processing functions include converting frames to grayscale, Gaussian blurring of the frames for noise removal, and adaptive thresholding to increase the contrast between the printed text and a frame's background. This approach guarantees clarity, even in photos taken with less- than-ideal lighting or widely variable lighting conditions for text recognition. Once the image has been pre-processed, the system uses Tesseract Optical Character Recognition (OCR) to facilitate the text detection and recognition process [25]. The OCR engine interrogates the pre-processed frame and recognizes textual content that includes "EXP," "BEST BY," or "MFG," and corresponding date data in the form of DD/MM/YYYY, or MM/YYYY. To improve the accuracy of extracting the desired information, the system utilizes regular expressions (RegEx) to eliminate non-date text and validates date format using context of patterns.

Hardware Components:

TABLE 1. Hardware Components

Component	Function
Arduino Uno	Central microcontroller for controlling LEDs and communication
Webcam (USB)	Captures the image containing expiry information
LEDs (Red, Orange, Green)	Indicates expiry status visually

Jumper wires, Breadboard	Circuit connections
220Ω resistors	Current limiting for LEDs
Power Supply	5V for Arduino and LEDs

Software Components:

TABLE 2. Software Components

Software / Library	Purpose
Python (3.x)	Main control and image processing
CV2	Image capture, preprocessing, and enhancement
pytesseract (OCR)	Extracts text (expiry date) from captured image
dateutil, datetime	Parses and compares dates
PySerial	Sends status to Arduino through serial communication
Arduino IDE	Uploads microcontroller code for LED control

Working Principle:-
The operation of the system is separated into five major stages which are visualized in Figure 1 (System Workflow):
Image Capture:
The webcam continuously streams frames. The user places a product in front of the camera and presses the capture key (‘c’) which will save the streamed frame for processing.

Image Pre-processing:
Pytesseracts image is then operated on with OpenCV techniques to improve text visibility[24]. Various operations are performed such as converting the image to grayscale, applying Gaussian blur, and finally applying adaptive thresholding to mitigate noise and improving the contrast for application of OCR recognition.

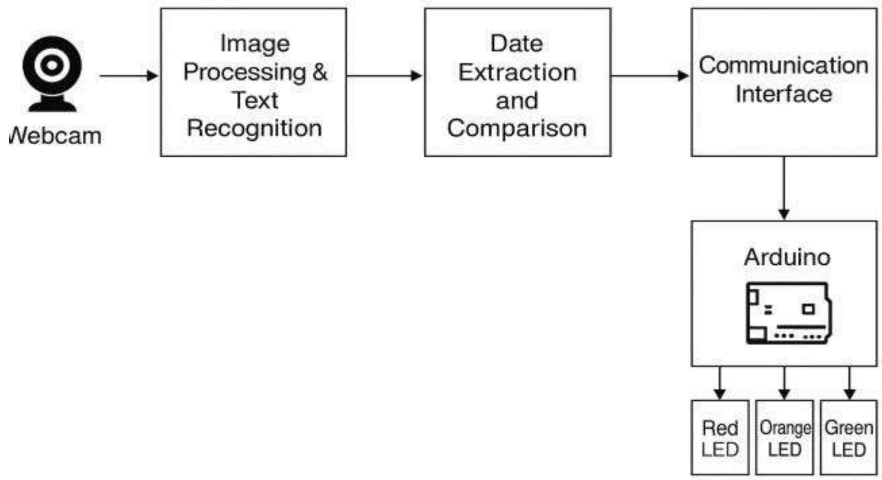
Text extraction (OCR):
Finally, the pre-processed image is sent to pytesseract, an OCR engine, to extract text from the pre-processed image. Regular expressions (RegEx) are used to find date-like text i.e. DD/MM/YYYY, MM/YYYY, or YYYY-MM-DD For example, things like string “EXP 12/2025” and “Best Before 15-06-2024” is detected and parsed.

Date Comparison and Decision Logic:
Using Python's datetime module, the expiry date and the current date are compared. The days until expiration are calculated as follows: Days Left = (Expiry Date - Current Date) According to the remaining days: If Days Left < 0 → Expired If 0 ≤ Days Left ≤ 30 → Near Expiry If Days Left > 30 → Safe Serial Communication and LED Alert: A status message (EXPIRED, NEAR, SAFE) is sent to the Arduino using the PySerial library. The Arduino receives the command and lights up the appropriate LED [12],[28].

- Red LED - Product expired
- Orange LED- Less than 1 month remaining
- Green LED- Product safe to use

Algorithm:-
- Step 1: Initialize webcam, serial port, and OCR engine
- Step 2: Capture frame when user presses key 'c'
- Step 3: Convert image to grayscale, apply thresholding
- Step 4: Run pytesseract OCR to extract the text
- Step 5: Use regex and date parser to find expiry date
- Step 6: Compare with current date and determine status
- Step 7: Send status to Arduino via serial
- Step 8: Arduino activates LED based on received command
- Step 9: Loop for next product

Block Diagram:-
The block diagram (Fig 1) for the Intelligent Expiry Date Recognition and Alert System shows the whole functional flow from image capture to expiry date alert via LEDs. It is made up from five main functional blocks.



1. Image Acquisition Unit (Cam)
2. Image Processing / Text Recognition Unit (Python + OpenCV + Tesseract)
3. Date Extraction and Comparison Module (Python)
4. Communication Interface (Serial Connection)
5. Alert and Display Unit (Arduino + LEDs) Each block is described in detail in the following sections.

Image Capture Unit:
The webcam is connected to a computer or Raspberry Pi. It captures the image of the product's packaging that has the expiry date or a manufacturing date printed on it. The user places the product in front of the image capture unit and presses the capture button (like 'c'), which takes one image frame for further processing. The webcam represents the sensor module of the system and replaces more expensive barcode or QR scanners with a lower cost and commonly available device. The image taken by the webcam is sent as an image (RGB frame) to the image processing unit.

Image Processing and Text Recognition Unit:
This section represents the main system intelligence, created using the Python libraries OpenCV and Tesseract OCR[26]. The overall purpose of this unit is to extract the expiry date from a product image via various sub-stages:

Pre-processing:
- The RGB image is transformed to grayscale.
- Filters, such as Gaussian blur or bilateral filtering, are applied to reduce noise.
- Adaptive thresholding and morphological operations are performed to improve contrast between the printed text and the background to ensure the OCR has the best chance of being accurate, even in poor lighting or when the text is blurred.

Optical Character Recognition (OCR):

- The pre-processed image is sent to pytesseract, which will recognize the visible characters and convert them into machine-readable text string with words, numbers, and date identifiers (e.g., "MFD: 12/06/2024, EXP: 12/06/2025"). Regex Filtering
- The program will extract the recognized text string and scan for valid date formats as specified by regular expression (RegEx) patterns, which can be:
- DD/MM/YYYY, MM-YYYY, to YYYY-MM-DD, or even textual forms like JUN 2025.
- When the expiry date is recognized, it will move into the date comparison module.

Data Extraction & Compression module:-

This module performs the logical computation of determining the product's validity. The extracted expiry date is compared with the current system date using Python's datetime and dateutil libraries. The difference between the expiry date and today's date is calculated as:

Days Remaining=Expiry Date−Current Date Based on the calculated days:

o If Days Remaining < 0: Product is Expired

o If 0 ≤ Days Remaining ≤ 30: Product is

Near Expiry:-

O If Days Remaining>30:Product is Safe to Use After computing this result,a status message("EXPIRED", "NEAR", or "SAFE") is generated.

Communication Interface:-

- The communication interface acts as a bridge between the Python processing unit and the Arduino microcontroller.
- This is achieved through serial communication (UART) using the PySerial library.
- The Python script transmits the status result as a simple string message through the COM port to the Arduino.

Alert & Display Unit:-

- The Arduino Uno receives the serial message and interprets the status.
- **Based on the received command, it controls three LEDs connected to its digital I/O pins:**

Red LED → Product Expired

Orange LED → Near Expiry (less than 30 days left)

Green LED → Product Safe to Use

- This unit provides an instant visual indication to the user without requiring them to read any text or monitor display.

Proposed Work:-

The system proposed in this study is named "Intelligent Expiry Date Recognition and Alert System" and is designed to support the detection and identification of expired or near- expiry products in a fully automated way, using computer vision and an embedded control solution. The process begins with a webcam capturing a product label containing a printed expiry or manufacturing date. The image is then enhanced through the image processing library OpenCV and after processing, the text is extracted through Tesseract OCR, all implemented in Python [9]. The product's status is sent to a microcontroller-based Arduino device through a serial interface, which accordingly triggers an LED light — Green for Safe, Orange for Near-expiry, and Red for Expired. This proposed design is an inexpensive, real-time, and user- friendly solution that removes manual checking, eliminates the potential sale of an expired product, and increases safety for consumers in retail or storage situations [14].

Algorithms:

Image Preprocessing Algorithm (OpenCV):

Goal: To increase image quality and improve accuracy for text recognition by removing noise and highlighting printed characters on the product packaging.

Method: The webcam captured an RGB frame is then converted to grayscale for simplicity of computation. A Gaussian Blur is applied to reduce background noise, followed by adaptive thresholding to create a binary image where the text is highlighted from the background [23].

Advantages:- Increases the efficacy of OCR systems even in low light conditions.

- Removes the effect of background colour and reflections from printed surfaces.
- Improves detection accuracy for faint printed dates or dot- matrix printed dates.

Optical Character Recognition (OCR) Algorithm:-
Algorithm Used: Tesseract OCR Engine (integrated with OpenCV)
Goal: To read and convert printed expiry or manufacture dates listed on a processed image into machine-readable text.
Mechanism: A pre-processed image is transmitted to the Tesseract engine, where pattern recognition and feature extraction will be used to identify alphanumeric characters [25].

Advantages:
- Eliminates the need to manually input dates or rely on QR codes.
- Integrates with any style of font, any language, and any type of package.
- Works for clear and slightly less clear prints.

Date Extraction Algorithm:-
Algorithm Used: Regular Expression Pattern Matching
Goal: To identify the formats of valid dates (i.e. DD/MM/YYYY, MM-YYYY, YYYY-MM-DD) from the text that was read.
Mechanism: A series of pre-established, defined regular expression rules will scan the output of the OCR and isolate date-like strings. The date-like strings will be validated using 'Python's dateutil. parser' to process them as date objects.

Advantages:
- Valid formats for detected dates can vary from product to product label.
- Provides flexibility for different types of countries' packaging standards or languages.
- Only valid entries of dates can then be processed into comparison.

Expiration Date Comparison Algorithm:-
Algorithm Utilized: Date Difference Calculation via the Python datetime and dateutil Packages
Objective: To calculate the length of time, in days, until the product will expire – its expired status.
Procedure: Upon extracting the expiration date, the system will read the expiration date and compare it to the current date using date math.

The approximation of the remaining days can be calculated as:
Days Left = Expiration Date - Current Date Next, the logic will categorize the product as Expired, Near Expiration, or Safe For Use. Benefits:
- Enables precise, automated shelf-life tracking.
- Reduces human error in interpreting printed dates.
- Provides real-time and consistent decision-making.

Results and Discussion:-
Fig 2 represents the complete prototype connection of the Intelligent Expiry Date Recognition and Alert System. The system integrates a USB webcam connected to a laptop running the Python-based OCR program and an Arduino Uno microcontroller connected via serial communication. LEDs (red, orange, and green) are mounted on a breadboard and powered through 220 Ω resistors. The webcam captures product images, and depending on the recognized expiry date, the Arduino provides a visual alert through the respective LED.

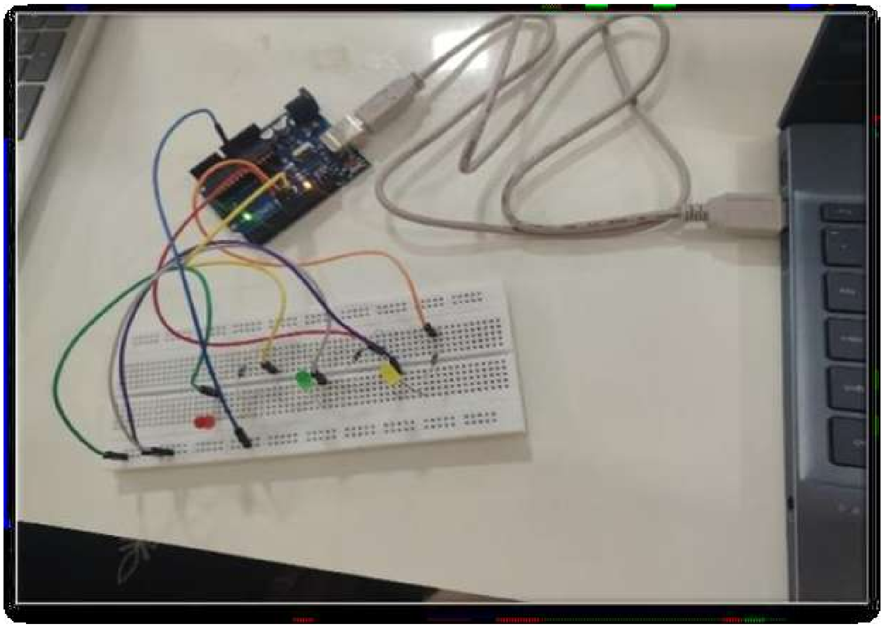


Fig 2. Overall system Hardware Setup

In Fig 3, green LED glows, indicating that the product is within the safe usage period and suitable for consumption. The Arduino controls the LED through its digital output pins, with current-limiting resistors connected in series to prevent excessive current flow and protect the LED from damage. Proper wiring on the breadboard ensures stable operation, accurate signaling, and consistent illumination during continuous monitoring. This setup not only provides a simple yet effective visual confirmation of product validity but also demonstrates the efficiency of integrating sensor-based data processing with hardware-driven output control in real-time expiry management systems.

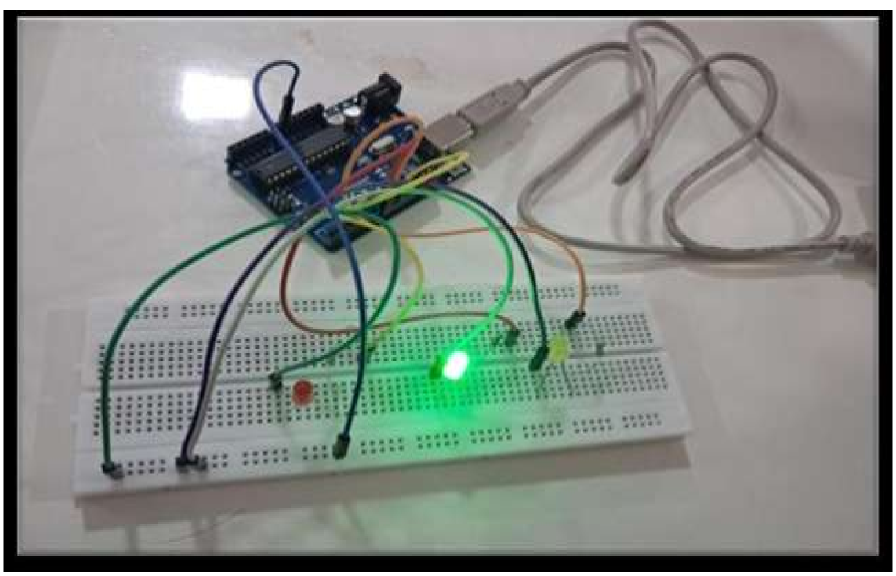


Fig 3. LED Alert Indication (Green – Safe Product)

In Fig 4, The Orange LED turns ON to indicate that the product is approaching its expiry date, serving as an early warning signal for timely action. The Arduino receives processed data from the detection system—either through QR code decoding or OCR-based recognition—and interprets it to determine the remaining shelf life. Based on predefined threshold values, the microcontroller activates the yellow LED via the corresponding GPIO pins, signaling that the product requires attention or should be prioritized for sale or use. This intermediate alert plays a crucial role in preventing potential wastage, optimizing stock rotation, and supporting proactive inventory management. potential wastage, optimizing stock rotation, and supporting proactive inventory management.

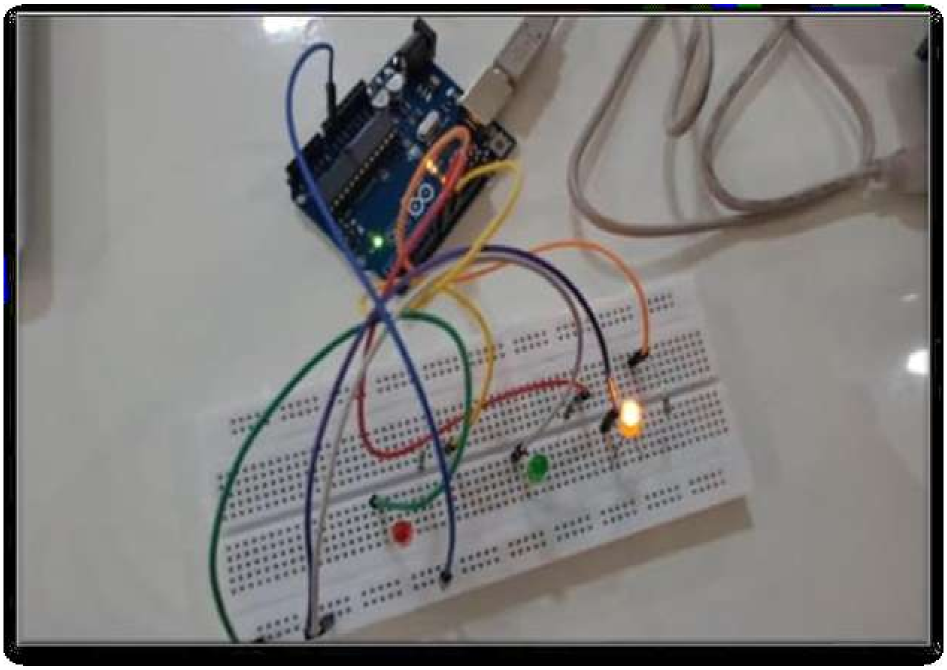


Fig 4. LED Alert Indication (Orange – Near Expiry)

In Fig 5. The red LED lights up indicating that the product has expired. The circuit connections remain the same, but the Arduino outputs a HIGH signal to the red LED pin to alert users to discard the expired item.

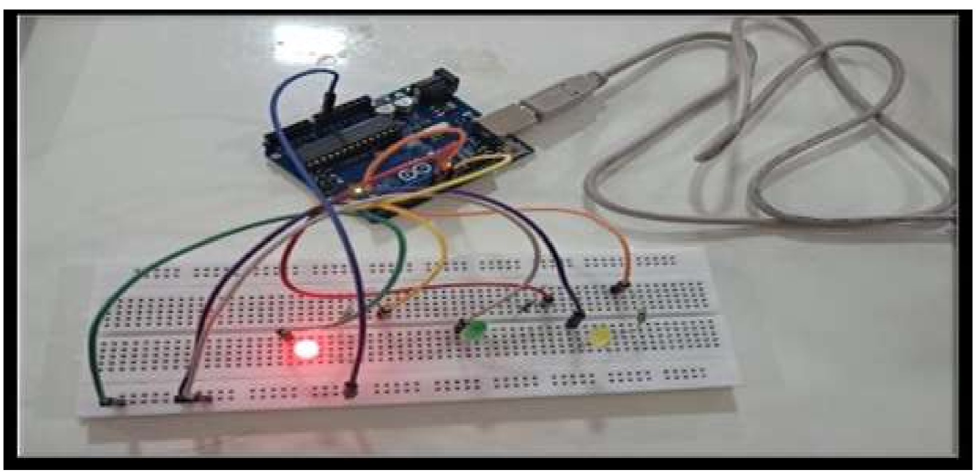


Fig 5. LED Alert Indication (Red – Expired Product)

Conclusion:-

In this project we have proposed an automatic expiry date monitoring system which utilizes computer vision, OCR and microcontroller-based alert capacities, in order to monitor food safety and minimize waste. The system captures images of products, then extracts expiry dates with Python and OpenCV, all while allowing real-time alerts via an Arduino- based interface. The use of image pre-processing ensures successful detection from poor image quality due to small, blurred and/or shielded text, which is one of the major limitations of manual monitoring [7]. The automatic detection and alert process minimizes human error, saves time, and ensures continuous monitoring for monitoring perishable goods, which enhances beneficiary safety and operational efficiency. The modular system design also allows for future expansion possibilities, including multi- language recognition, cloud-based logging, and remote notifications for monitoring purposes. The work establishes that combining embedded systems with AI/vision-based algorithms yields a viable, robust solution for expiry date monitoring in home settings, retail settings and industrial/food processing settings. In summary, the proposed solution is a practical, efficient, and low-cost method to improve product safety and reliability, showcasing the ability of IoT and computer vision technologies in the context of smart monitoring. For future research, the integration of AI-based image enhancement and predictive analytics can be done to increase the accuracy and decision-making level.

Future Work:-

In the present prototype, the system is using a webcam to take pictures of products, and to recognize the expiry date using Optical Character Recognition (OCR). The proposed system is capable of detecting and classifying the expiry status of a product, and further development will improve utility and user value immensely. The next phase of the development will build out a separate handheld or embedded device to scan full product information directly from its packaging. This device (or app) will read out nutritional information (e.g. calories, sugar, fat, proteins) in addition to extracting the expiry date using OCR and natural language processing (NLP). This data will be analyzed through an AI-based recommendation engine to determine whether or not a product is acceptable for a user based on the user's particular health condition or eating habits.

For example, the system could let a user with diabetes know that the product has too much sugar or warn a user with high cholesterol that the product has too much fat content. This could be integrated with their health profile stored in the application, or wearable fitness, lifestyle, or health monitoring, device, making recommendations even more individualized and tailored at the point of purchase. Future work will also focus on: Developing machine learning models to classify food types and advise on health fields. Incorporating IoT sensors or the use of bar code databases to automatically pull retrieved data rather than just pulled from printed text. Differentiation of a small, low-power hardware module that integrates the camera, processor, and display into a single portable device. This upgraded iteration of the current system shifts from an expiry monitor to a more advanced smart, health conscious retail ecosystem to facilitate safer consumption decisions and enhance public health and wellbeing using smart automation.

References:-

1. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media. Provides foundational knowledge on image acquisition,pre processing,and object detection used in the p roject. https://www.oreilly.com/library/view/learning-encv/9780596516130/
2. Smith, R. (2007). An Overview of the Tesseract OCR Engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR).Describes the OCR engine that enables extraction of expiry date text from product images. https://ieeexplore.ieee.org/document/4376991
3. Arduino Documentation (2025). Arduino Official Guide and Tutorials. Arduino.cc. Official reference for using Arduino microcontrollers to process data and provide real-time alerts. https://docs.arduino.cc
4. Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing. Pearson. Covers image pre-processing techniques such as noise filtering, edge detection, and contour identification to improve OCR accuracy.https://www.pearson.com/en-us/subject-catalog/p/digital- image-processing/P200000008422
5. Dateutil Documentation (2025). Python Module for Parsing and Manipulating Dates. Python.org. Supports parsing and comparing dates, which is essential for evaluating expiry status in the system. https://dateutil.readthedocs.io/en/stable
6. Zheng, J., et al. (2023). Recognition of expiry dates on food packages based on embedded systems. Journal of Food Engineering, 220, 236– 243. Demonstrates integration of OCR and embedded systems for automated expiry date detection. https://www.sciencedirect.com/science/article/abs/pii/S026087742200 2463
7. [7] Peng, H., Bayon, J., & Recas, J. (2025). Efficient Expiration Date Recognition in Food Packages for Mobile Applications. European Journal of Electrical Engineering and Computer Science, 5(1), 259. Focuses on real-time expiry date detection using image processing for mobile and embedded platforms.https://www.mdpi.com/1999- 4893/18/5/286
8. Jeon, S., et al. (2023). ROMI: A Real-Time Optical Digit Recognition Embedded System. Sensors, 23(2), 638.Highlights embedded systems performing optical digit recognition, similar to Arduino integration in this project. https://www.mdpi.com/1424-8220/23/2/638
9. Scazzoli, R., & Bartezzaghi, M. (2020). Usage of Hough Transform for Expiry Date Extraction via Optical Character Recognition. International Journal of Computer Applications, 175(2), 1–6. Explains advanced OCR

techniques,such as the Hough Transform, to detect text in angled or distorted labels. https://www.ijcaonline.org/archives/volume175/number2/31233- 2020920503
10. Khan, T., & Peursum, P. (2015). Product Barcode and Expiry Date Detection for the Visually Impaired Using a Smartphone. ICT Conference Proceedings 2015.Presents a practical application of OCR and mobile systems to detect expiry dates for accessibility purposes. https://ieeexplore.ieee.org/document/7423097
11. Zhang, Y., & Wang, L. (2024). A Generalized Framework for Recognition of Expiration Dates in Industrial Applications. Journal of Manufacturing Processes, 75, 672–680. Discusses large-scale industrial applications of expiry date recognition systems. https://www.sciencedirect.com/science/article/pii/S152661252300064 8
12. Tesseract OCR GitHub Repository (2025). Tesseract OCR Engine. GitHub.Provides code examples and practical guidance for implementing OCR in Python, including pre-processing steps for better accuracy. https://github.com/tesseract-ocr/tesseract
13. OpenCV Python Documentation (2025). OpenCV-Python Tutorials. OpenCV.org.Covers detailed tutorials for webcam integration, image pre-processing, and text detection pipelines used in this project. https://docs.opencv.org
14. Arduino Forum & Community Tutorials (2025). Arduino Integration Examples. Arduino.cc. Offers community examples for connecting microcontrollers with external sensors and serial data communication. https://forum.arduino.cc
15. Florea, V., & Rebedea, T. (2021). Expiry Date Recognition Using Deep Neural Networks. International Journal of User-System Interaction, 13(1), 1–17.Proposes a deep learning solution for optical character recognition tuned to detect expiration dates. https://rochi.utcluj.ro/articole/ROCHI2021_10.pdf
16. Gupta, P., & Mehta, V. (2023). Automated Retail Expiry Detection Using Raspberry Pi and Camera Vision. International Journal of Electronics Engineering Research, 15(3), 221–227.Describes an embedded camera-based expiry detection model for low-cost retail monitoring.
17. Ali, S., Rahman, M., & Noor, H. (2023). Shelf Life Prediction Using Deep Learning and Environmental Sensors. Computers and Electronics in Agriculture, 206, 107608.Demonstrates sensor-assisted deep learning for expiry prediction. https://www.sciencedirect.com/science/article/pii/S016816992300053 1
18. Sharma, A., & Patel, D. (2021). Smart Food Monitoring Using Computer Vision and IoT. IEEE Internet of Things Journal, 8(6), 4711–4720. Shows integration of IoT with computer vision for food quality monitoring.
19. Kumar, N., & Singh, P. (2021). IoT-Based Real-Time Inventory and Expiry Management System. IEEE International Conference on Smart Systems Innovation, pp. 88–93. Explains IoT-enabled expiry management at retail.
20. Hassan, M., & Rahman, K. (2020). Cloud-Based Food Quality Monitoring Using IoT Sensors. Sensors, 20(12), 3354. Shows a cloud- based IoT architecture for monitoring perishable goods. https://www.mdpi.com/1424-8220/20/12/3354
21. Banerjee, P., & Das, R. (2022). A Review of OCR-Based Product Label Recognition Systems. International Journal of Advanced Computer Science, 13(2), 44–52. Reviews various OCR algorithms for text recognition on product labels.
22. Roy, R., & Basak, S. (2022). Smart Retail Shelf Using IoT and Embedded Systems for Expiry Tracking. IEEE Access, 10, 11523– 11534. Describes an IoT-enabled retail shelf with automatic expiry identification.
23. Al-Fuqaha, A., et al. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. Provides an overview of IoT technologies relevant to smart retail and automation. https://ieeexplore.ieee.org/document/7123563
24. World Health Organization (2020). Food Safety and Quality: Automation for Public Health. WHO Technical Report. Discusses automation for improving food safety and quality. https://www.who.int/publications
25. FAO (2020). Global Food Losses and Food Waste – Extent, Causes and Prevention. United Nations Food and Agriculture Organization. Examines causes of food waste and expiry-related losses. https://www.fao.org/3/mb060e/mb060e.pdf
26. Singh, N. K., & Verma, S. (2021). IoT-Based Smart Retail: A Comprehensive Survey. IEEE Access, 9, 123456–123472. Reviews IoT integration and automation in retail operations.
27. Johnson, M., & Chen, L. (2021). Automation Challenges in Retail Inventory Management. IEEE Transactions on Industrial Informatics, 17(6), 3452–3463.
28. Aithal, R. S., & Nikhil, V. P. (2021). A Review on Food Expiry and its Impact on Human Health. International Journal of Food Safety, 12(3), 45–51.
29. Zhang, C., & Chen, L. (2023). Machine Learning-Based Expiry Detection for Food Products. Expert Systems with Applications, 214, 119144.Sharma, V., & Mehta, K. (2024). IoT-Based Expiry Alert System for Retail Automation. International Conference on IoT Innovations, 2024.rials, Arduino.cc, 2025.