Smart Tray: An IoT and Object Detection-Based Solution for Automated Household Grocery Management

Muhammed Shahil A K[1], Naswiha M K[2]
[1]Department of Computer Science, Sullamussalam Science College, Areekode
Email: mdshahilak@gmail.com
[2]Department of Computer Science, Sullamussalam Science College, Areekode
Email: naziameen9797@gmail.com

Abstract

This article describes the creation of a "Smart Tray" prototype that uses IoT and object recognition to improve supermarket inventory management for busy homes. The Smart Tray detects and counts fruits and vegetables placed on it, allowing for real-time tracking of shopping stock. When the count of any item goes below a certain threshold, the system notifies the user, ensuring immediate restocking. Also, the prototype allows customers to place orders directly with shops via an integrated interface, which increases convenience.The motivation for this initiative comes from the usual issues that working families experience when managing and keeping household food. The Smart Tray automates inventory checks, reducing human labor, lowering the chance of running out of essential items, and encouraging effective home management. Technologically, the system uses IoT devices and the MobileNet SSD (Single Shot Detector) model to reliably recognize and categorize fruits and vegetables. A custom dataset is used to train the model, allowing for real-time item recognition and tracking.This study also looks at the larger implications of IoT-based inventory systems, such as their potential use in smart homes and retail contexts. Future goals include publishing this study to help advance academic and industrial understanding.

Keyword :
Smart Tray, IoT, Object recognition, Prototype, MobileNet SSD, Custom dataset

1. Introduction

In the modern era, technology continues to transform everyday life, making tasks more efficient and convenient. Household inventory management is one important area where innovation can have a big influence. Families frequently find it difficult to keep an eye on their grocery supplies due to busy schedules and demanding lifestyles, which can result in instances where necessary items run out of stock without warning. The idea of a "Smart Tray"—an Internet of Things-based device that combines object identification and real-time tracking to automate the supermarket management process—was created in order to address this difficulty.

The Smart Tray represents a fusion of Internet of Things (IoT) technology and Artificial Intelligence (AI), designed to cater to the needs of busy households. The Smart Tray uses object detection algorithms to accurately detect and count food items like fruits and vegetables in real time, utilizing IoT-enabled devices. To ensure prompt restocking, the system notifies the user's mobile device when the amount of an item drops below a predetermined threshold. The tray also makes it possible to place orders with store owners directly, which expedites the process of restocking necessary supplies.

SSD MobileNetV3 model, trained on a modified version of the Fruit-360 dataset, is at the heart of the Smart Tray. Real-time, exact fruit and vegetable identification and categorization are made possible by our lightweight deep learning model. The system's components are processed and integrated easily due to the hardware configuration, which is based on the small and powerful Raspberry Pi 5. The Smart Tray provides an example of how new technologies may improve and automate household tasks when used in combination with Python-based software and tools such as PyTorch and OpenCV.

The increasing necessity for automation in daily life is the motivation behind this initiative. Particularly for working families, traditional supermarket management techniques that require manual checks and repeated trips to the shop can be ineffective and time-consuming. The Smart Tray aims to solve these inefficiencies by offering a practical, automated solution that improves household management, minimizes human labor, and avoids stockouts.

The Smart Tray idea has greater potential for smart homes system in addition to its immediate usage in houses. This research demonstrates how technology may enhance inventory management, reduce down on food waste, and enhance decision-making by combining IoT with object identification. The Smart Tray, as a prototype, also provides a basis for further advancements in the field, offering chances for improved scalability, functionality, and commercial applications.

The Smart Tray's methodology and implementation are described in this document, with a focus on the integration of hardware and software components, object detection model training and deployment, and system testing under real-world situations. The findings show the revolutionary impact of AI and IoT in daily life while attempting to advance academic and industry understanding of IoT-based inventory management solutions.

2. Literature Review

This paper by Faisal Mehmood, Israr Ullah, Shabir Ahmad, DoHyeun Kim presents a smart home automation system using the Single Shot Detector (SSD) algorithm for object detection on IoT-enabled embedded devices, with AWS cloud-based control and monitoring. Using MQTT protocol for communication, the system employs Raspberry Pi and camera to detect objects and evaluate performance under varying conditions. Results show that environmental changes minimally impact processing delay, though lighting and frame size affect accuracy. A distributed broker design improves load management. This research demonstrates effective deployment of deep learning in real-time IoT applications, with insights into optimizing object detection for smart home environments. [1].

The paper by Horea Muresan and Mihai Oltean introduces the Fruits-360 dataset, a high-quality collection of over 90,000 images of 131 types of fruits and vegetables, aimed at improving fruit classification through deep learning. The dataset minimizes background noise for accurate object recognition, which is crucial in various applications like autonomous robots, augmented reality, and fruit harvesting. The authors train a neural network using TensorFlow to classify a range of fruits, targeting use in complex scenarios like autonomous store inspections and agricultural automation. The paper also outlines the network's architecture, performance results, and future improvements for broader applicability. [2].

The work by Muhammed Shahil A K, Jalwa V P, Afnan MK, Rababa Kareem K, Muneer V K introduces an "Automated Catalogue System using Object Detection" to enhance security and wildlife monitoring through real-time object recognition. Employing technologies like SSD, OpenCV, and MobileNetV3, the system categorizes detected objects, capturing images and recording details like category, time, date, and confidence level. Designed to address challenges in surveillance and animal intrusion into human areas, the system provides a user-friendly web application for accessible records. By combining accurate categorization, image capture, and real-time detection, this system offers a versatile tool for security and wildlife management, supporting both community safety and environmental awareness. [3].

The paper by Sudharshan Duth P, Jayasimha K presents a deep learning-based system for recognizing vegetables, addressing challenges in distinguishing visually similar varieties (e.g., red tomato vs. red capsicum) by using convolutional neural networks (CNNs). Traditional methods relying on color, texture, and shape features often lead to misclassification due to vegetable color variations across ripeness stages. This research achieves an efficient intraclass vegetable recognition system with 95.5% accuracy by leveraging CNNs to learn complex image patterns, enhancing the precision of vegetable detection. The study uses 24 vegetable types and highlights CNN’s superiority over feature-based methods, advancing applications in computer vision for accurate food identification. [4].

This study by Manya Afonso, Hubert Fonteijn, Felipe Schadeck Fiorentin, Dick Lensink, Marcel Mooij, Nanne Faber, Gerrit Polder and Ron Wehrens explores the use of MaskRCNN for accurately detecting and counting tomatoes in greenhouse images, addressing labor-intensive manual phenotyping and advancing automation in agriculture. MaskRCNN’s object detection and segmentation capabilities demonstrate reliable results in greenhouse conditions, comparable to or better than those achieved in controlled lab settings. The approach efficiently manages challenges such as varying lighting, colors, and plant positioning, while also implicitly learning object depth. This method contributes to the broader automation of crop monitoring, benefiting yield prediction and easing labor demands in horticulture [5].

The paper by Sudharshan Duth P, Jayasimha K presents an efficient system for classifying and counting fruits and vegetables at checkout counters without barcodes. Using a hybrid model that combines EfficientNet for classification and a Decision Tree for counting based on weight, this solution achieves 80% counting accuracy and rapid processing (0.2 seconds per image) on a CPU. The model operates without costly GPUs, reducing hardware requirements while maintaining speed and accuracy. It supports real-time item recognition and inventory monitoring, enhancing cashier efficiency and customer satisfaction by minimizing errors and long wait times. [6].

This paper introduces a deep learning model for automatic fruit yield estimation using a modified Inception-ResNet architecture trained on synthetic data. Aiming to support farmers in decision-making by accurately counting fruits even in challenging conditions like shadow, occlusion, and overlapping, this approach achieves 91% accuracy on real images. By using simulated data, the method eliminates the costly need for large, labelled datasets. The model’s efficient, real-time performance demonstrates its suitability for robotic agricultural applications, providing a practical solution to traditional, labor-intensive counting processes [7].

Islam et al. classified food photos using a CNN that was created from design using the FOOD-11 dataset.Eleven food categories were taken into consideration, including dairy, bread, meat, eggs, soup, and products. noodles, fried meals, rice, desserts, and fruits. During image pre-processing, ZCA whitening was used to cut down on redundancy. SGD and Adam optimizer. The photos were classified using optimizers. The accuracy of the model was 74.70%. For comparison, they also employed the Inception v3 model, which has already been trained on the ImageNet dataset. The accuracy of Inception v3 was 92.86% [8].

Jian et al. successfully integrated deep learning and machine vision. They developed R-FCN, a deep learning technique that combined an area proposal network and a completely convolutional neural network for fruit detection and localization. The proposed technique extracted pixel-level information by convolving the input image with FCN. Combining remaining networks gave the deep network more feature information to recognize the fruit, and deconvolution made it possible to display the detection findings. Following convolution, RPN produced a significant number of boxes on the feature area. It effectively distinguished between the top and bottom regions of the image. Three distinct fruits from various states were used for testing, and the public COCO data set was used for training. According to the trial results, the method used in this work enhances detection accuracy by 0.71% and 0.33% when compared to the previous system that distinguished between apples and oranges. Additionally, it had an accuracy of 82.3% in identifying bananas, a fruit that is cultivated on large amounts]. By visualizing fruit position and detection over a range of input images, it reduces the effect of branch and leaf blocking, increases picking efficiency, and strengthens the system. Their work was limited by the system's increased categorization processing time [9].

For deep feature extraction, Şengür et al. chose pre-trained AlexNet and VGGl6 models. Properties of size 4096 were extracted from the models' fc6 and fc7 layers. The best deep feature sequence for categorizing the food image was then created by combining these features in various combinations. SVM was then applied to classify the combined attributes. The proposed method was tested using publicly accessible datasets, FOOD-5K, FOOD-101, and FOOD-11, and performance was evaluated using the accuracy metric [9].The accuracy of the FOOD-5K dataset was 99.00%, while the accuracy of the FOOD-11 and FOOD-101 datasets was 88.08% and 62.44%, respectively. Also, they improved the CNN model, which was already trained on the FOOD-101 dataset and had a 79.86% accuracy rating. The obtained results were compared with a number of different techniques on the FOOD-11 and FOOD-101 datasets. It was discovered that the proposed method outperformed the others.[10].

3. Methodology

The methodology for developing the "Smart Tray" prototype was structured in phases, each addressing key technical components for a easy integration of IoT, object detection, and real-time notifications. The first phase involved selecting appropriate hardware, including the Raspberry Pi 5, which serves as the core processing unit. The second phase focused on data collection and model training, where the Fruit-360 dataset was used to train an SSD MobileNetV3 model to accurately identify and classify various fruits and vegetables. Next, real-time object detection was implemented using OpenCV and PyTorch on the Raspberry Pi, enabling the system to monitor and count items in the tray. The final phase integrated an alert system, sending notifications to the user when item quantities fall below a set threshold, allowing users to place orders through a mobile interface.

3.1 Hardware Setup

The hardware setup aimed to provide a robust platform for the Smart Tray's operations. A Raspberry Pi 5 served as the primary processing unit, chosen for its compact size and sufficient computational power for running lightweight deep-learning models. A camera module was installed to capture high-resolution images of the tray's contents. The tray itself was constructed using a sturdy base to hold fruits and vegetables securely. Powering the Raspberry Pi was a stable and reliable power source, ensuring uninterrupted functionality. The hardware components were selected for their compatibility, cost-effectiveness, and suitability for real-time use cases.

3.2 Software Development

The software backbone of the Smart Tray was implemented in Python, leveraging its rich ecosystem of libraries and frameworks. Key technologies included PyTorch for deep learning model deployment, OpenCV for image processing tasks, and MQTT for IoT communication. The system was coded to perform multiple tasks, including image capture, preprocessing, and feeding the processed images to the object detection model. Additional features included a notification system that alerts users when an item's count falls below a threshold and an interface for placing orders directly from a mobile device.

3.3 Single Shot Multibox Detectors(SSDs)

A Single Shot Detector (SSD) is an innovative object detection technique for computer vision. Its ability to quickly and accurately recognize and find elements within image or video frames sets it apart. SSD is unusual in that it can accomplish this in a single run of a deep neural network, making it highly effective and ideal for real-time applications. SSD does this by utilizing several anchor boxes in feature maps with varying aspect ratios. It can successfully manage items of varied sizes and shapes thanks to these anchor boxes. Also, SSD employs multi-scale feature maps to detect items at various sizes, ensuring accurate recognition of both large and tiny objects in the picture. SSD is a useful tool for work requiring many item categories in a single image because of its capacity to detect numerous object classes at once. The SSD architecture is illustrated in Figure 2 below.

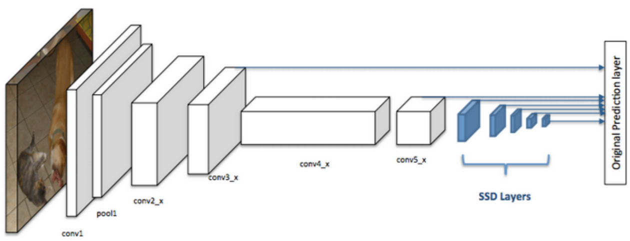

Fig 2. SSDs Architecture.

3.4 MobileNet

MobileNet is a CNN architectural model developed for image classification and mobile vision. While there are various models, MobileNet stands out since it requires very little computing resources for transfer learning applications. As a result, it works well with mobile devices, embedded systems, and PCs with poor computational efficiency or no GPU, all while retaining substantial output accuracy.

3.4.1 MobileNetV3

MobileNetV3 is a more advanced and efficient way to build neural networks for applications like image recognition. MobileNetV3 determines the ideal network design without the need for human involvement with AutoML, a type of machine learning. It uses a mix of MnasNet and NetAdapt algorithms to generate an approximation design, which is eventually optimized for maximum performance. MobileNetV3's core design includes "squeeze-and-excitation" blocks, which is one of its characteristics. These blocks focus on the most important components while ignoring minor details, allowing the network to obtain higher-quality data. This model was trained using the Coco dataset. This increases the network's capacity to understand and recognize objects. Figure 3 shows the MobileNetV3 block underneath.

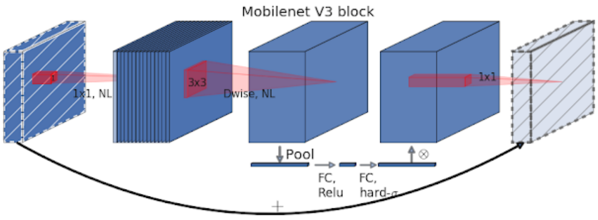

Fig 3. Block of MobileNetV3

Another innovative element of MobileNetV3 is how it optimizes some of its design's more complicated components. This eliminates three difficult layers while maintaining accuracy and enhancing network performance. In trials, MobileNetV3 showed a 25% decrease in object identification time compared to earlier versions, while keeping the same level of accuracy. MobileNetV3 is a more intelligent and cost-effective approach of developing neural networks for applications such as object detection in photographs. Optimisations, automatic learning, and smart design decisions are used to make the network quicker and smarter.

3.5 Dataset

The Fruit-360 dataset was selected as the primary data source for training the object detection model. It contains images of 24 distinct fruit and vegetable classes. Preprocessing the dataset involved resizing all images to 224x224 pixels, normalizing pixel values to match the input requirements of the SSD MobileNetV3 model, and applying data augmentation techniques such as flipping, rotation, and brightness adjustments. These preprocessing steps ensured the model's ability to generalize across diverse conditions, improving its robustness in real-world scenarios.

3.6 Model Training

The core detection mechanism was built using the SSD MobileNetV3 model. Transfer learning was employed by fine-tuning pre-trained weights to adapt the model to the Fruit-360 dataset. Key training parameters included a learning rate of 0.001, a batch size of 32, and 50 training epochs. The model was trained using a supervised learning approach, and metrics such as accuracy, precision, recall, and F1-score were used to monitor its performance. The trained model demonstrated high accuracy in identifying and categorizing fruits and vegetables.

3.7 Real-Time Detection and Notification

After training, the model was deployed on the Raspberry Pi to enable real-time object detection. The camera module continuously captured images of the tray's contents, which were processed using OpenCV for preprocessing and fed to the detection model for inference. The model identified the objects, counted them, and compared the counts with predefined threshold values. If an item's count fell below the threshold, the system sent an alert to the user via a mobile notification, using an MQTT-based messaging service. The tray also offered an interface for users to place grocery orders directly with shopkeepers.

3.8 Testing and Validation

The validation and testing phases revealed both strengths and limitations in the model's performance. While the model achieved an impressive overall validation accuracy of 100% in many epochs and a testing accuracy of 99.97%, the results also highlighted areas of concern. Notably, there were instances of overfitting, as evidenced by significant spikes in validation loss during certain epochs, such as Epoch 12 (validation loss: 8.0456) and Epoch 26 (validation loss: 16.9590). These fluctuations indicate the model's reliance on specific training data patterns rather than learning robust, generalizable features. Additionally, despite the high test accuracy, minor misclassifications were observed, especially in closely related classes like apple variants. These shortcomings emphasize the need for further optimization, including techniques like data augmentation, dropout regularization, and improved dataset diversity, to ensure consistent performance across diverse real-world scenarios.

3.9 Challenges and Future Enhancements

While the Smart Tray demonstrated success in its core functionalities, it faced limitations in distinguishing between visually similar items and maintaining high accuracy in poor lighting conditions. Future enhancements include expanding the dataset to incorporate additional classes, optimizing the model for faster inference on edge devices, and adding advanced IoT features like inventory analytics and predictive restocking. These improvements aim to increase the system's efficiency and broaden its applicability.

4. Detection Performance

The detection performance of the model demonstrates high accuracy on the test dataset, achieving an overall classification accuracy of 99.97%. However, this impressive result may not fully translate to real-world scenarios due to certain limitations. While the model performed exceptionally well on standard conditions, issues such as misclassifications in closely related categories (e.g., apple variants like apple_red_2) and reduced accuracy in low-light environments suggest challenges in generalization. Spikes in validation loss during training and fluctuations in accuracy further hint at overfitting, where the model has learned to perform well on training and testing datasets but struggles with unseen data or varied conditions. These observations indicate a need for enhanced dataset diversity and techniques such as data augmentation and regularization to improve detection reliability and robustness in real-world applications.

5. Conclusion

The "Smart Tray" project has shown how IoT and AI can work together to make grocery management smarter and more convenient. By combining object detection, real-time notifications, and an easy-to-use interface, the system helps track fruits and vegetables effortlessly. During testing, the model performed impressively, identifying most items with high accuracy and achieving a near-perfect testing score of 99.97%. However, the journey wasn't without its challenges. The training process showed signs of overfitting, and there were occasional errors in identifying similar-looking items. These issues remind us that there's always room to improve, such as using more diverse data and refining the model to handle real-world complexities better.

Despite these challenges, the Smart Tray has proven its value. It can detect and monitor items, alert users when stocks are low, and make life easier for busy families. Looking ahead, the system could be enhanced with features like connecting to online stores, recognizing a wider range of items, and improving energy efficiency. This project is just the beginning—a glimpse into how smart home technology can simplify everyday tasks and make our lives more seamless.

6. Reference

[1] Faisal Mehmood, Israr Ullah, Shabir Ahmad, DoHyeun Kim *"Object detection mechanism based on deep learning algorithm using embedded IoT devices for smart home appliances control in CoT"* Journal of Ambient Intelligence and Humanized Computing, Springer.
DOI:

[2] Horea Muresan, Mihai Oltean *"Fruit recognition from images using deep learning"* Computer Vision and Pattern Recognition, Acta Univ. Sapientiae, Informatica, 10, 1 (2018) 26–42
DOI:

[3] Muhammed Shahil A K, Jalwa V P, Afnan MK, Rababa Kareem K, Muneer V K *"Automated Catalogue System using Object Detection"* International Journal of Advanced Networking and Applications
DOI:

[4] Sudharshan Duth P, Jayasimha K *"Intra Class Vegetable Recognition System using Deep Learning"* IEEE Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020)
DOI:

[5] Manya Afonso, Hubert Fonteijn, Felipe Schadeck Fiorentin, Dick Lensink, Marcel Mooij, Nanne Faber, Gerrit Polder and Ron Wehrens *"Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning"* Frontiers in Plant Science November 2020 Volume 11 Article 571299
DOI:

[6] Sudharshan Duth P, Jayasimha K *"Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning"* ICAAI 2021, November 20–22, 2021, Virtual Event, United Kingdom
DOI:

[7] MaryamRahnemoonfar and Clay Sheppard "Deep Count: Fruit Counting Based on Deep Simulated Learning" www.mdpi.com/journal Sensors 2017
DOI:

[8] M. T. Islam, B. M. N. Karim Siddique, S. Rahman, and T. Jabid, *"Food image classification with convolutional neural network,"* in 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2018.
DOI:

[9] L. Jian, Z. Mingrui, and G. Xifeng, *"A fruit detection algorithm based on R-FCN in natural scene,"* in 2020 Chinese Control And Decision Conference (CCDC), 2020.
DOI:

[10] A. Sengur, Y. Akbulut, and U. Budak, *"Food image classification with deep features,"* in 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019.