Snaplist:Visual Shopping List Creator

D Sai Siva Bhaswanth
Artificial Intelligence and Machine learning
(VTU Affiliated)
RV college of Engineering
(VTU Affiliated)
Bengaluru,India
dssbhaswanth.ai22@rvce.edu.in

Chillale Naveen
Artificial Intelligence and Machine learning
(VTU Affiliated)
RV college of Engineering
(VTU Affiliated)
Bengaluru,India
chillalenaveen.ai22@rvce.edu.in

Kota Vishnu Datta
Artificial Intelligence and Machine learning
(VTU Affiliated)
RV college of Engineering
(VTU Affiliated)
Bengaluru,India
kvishnudatta.ai22@rvce.edu.in

Abstract— **The retail industry is witnessing rapid advancements with artificial intelligence and deep learning, enabling automation and personalized shopping experiences. Traditional shopping list applications require users to manually enter items, making the process inefficient and error-prone. SnapList introduces an AI-driven solution that automates shopping list creation using computer vision and deep learning. By capturing images of grocery items, users can generate an organized shopping list without manual input. MobileNetV3, an optimized lightweight deep learning model, ensures high accuracy in real-time object detection. The proposed system is designed for mobile environments, using TensorFlow Lite to reduce computational overhead while maintaining high performance. Experimental results demonstrate that SnapList achieves high classification accuracy and fast inference, making it a viable solution for modern retail applications. This research highlights the impact of AI in automating consumer tasks and optimizing shopping experiences.**

Keywords— computer vision, deep learning, shopping list automation, object detection, MobileNetV3, retail automation.

I. INTRODUCTION

A. Background and Motivation

Artificial intelligence and deep learning are revolutionizing the retail industry, enabling businesses to automate processes and enhance customer experiences. One area that has remained largely manual is shopping list management, where users rely on handwritten notes or digital applications requiring manual input. While digital solutions provide convenience, the need for manual entry makes them inefficient and prone to human errors.

Barcode-based shopping applications have been developed to simplify the process, allowing users to scan product barcodes to add items to their lists. However, these solutions have limitations, particularly when dealing with fresh produce, which lacks barcodes. Additionally, barcode scanners require clear visibility of product labels, making them unreliable in certain situations.

To address these challenges, SnapList introduces an AI-powered solution that automates shopping list creation through image recognition. By leveraging computer vision, the system allows users to generate grocery lists by simply capturing images of items, eliminating the need for manual input. The proposed solution integrates deep learning techniques to ensure high detection accuracy and efficient real-time processing on mobile devices.

B. Problem Statement

Traditional shopping list applications require users to manually enter grocery items, a process that is both tedious and prone to errors. While barcode-based solutions partially address this issue, they remain ineffective for products without labels or for scenarios where barcodes are not easily accessible. Furthermore, existing solutions do not offer real-time automation, leading to inefficiencies in shopping list creation.

SnapList aims to overcome these limitations by introducing a visual-based grocery list creator. By employing deep learning techniques, the system can recognize grocery items through image capture, providing an automated, efficient, and user-friendly shopping experience.

C. Objectives

The primary objectives of this research are:

1. To develop an AI-based system capable of automating shopping list creation using image recognition.
2. To implement MobileNetV3 for lightweight and efficient grocery item detection.
3. To enable real-time grocery recognition while minimizing computational overhead.
4. To deploy the system on mobile devices, ensuring accessibility and ease of use.

D. Contributions

This research contributes to AI-driven retail automation by:

1. Proposing a computer vision-based shopping list creator that eliminates the need for manual entry.
2. Demonstrating the use of MobileNetV3 for efficient and accurate grocery detection.
3. Evaluating the system's performance on a large-scale grocery dataset to validate its effectiveness.

II. LITERATURE REVIEW

A. Existing Work in Shopping List Automation

Several studies have explored deep learning applications for grocery item recognition in retail environments. Patel and Gupta propose a convolutional neural network-based approach to automate product identification. Their model, trained on a diverse supermarket dataset, applies data augmentation techniques such as rotation and brightness adjustments to improve classification accuracy. Achieving an accuracy of 94.7 percent, their research highlights the potential of CNNs to replace barcode scanning, streamlining the checkout process and improving shopping efficiency.

Lee and Kim introduce a hybrid model that combines convolutional neural networks with Vision Transformers to enhance grocery item recognition. Conventional CNN models struggle in cluttered retail environments due to occlusions and similar-looking products. By incorporating self-attention mechanisms, their model better distinguishes between product features, improving accuracy by 7.3 percent over standard CNNs.

Their research emphasizes the importance of minimizing background noise and improving feature extraction for efficient product identification.

Sharma and Mehta explore YOLOv5 for real-time grocery detection, introducing a shopping assistant capable of recognizing multiple items in a single frame. Their system, integrated into a mobile application, automatically adds detected items to a shopping list. With an inference speed of 15 milliseconds per image, YOLOv5 outperforms Faster R-CNN and SSD models, making it suitable for low-latency applications. Their research highlights the benefits of AI-driven shopping assistants for accessibility and contactless shopping.

Nguyen and Lin investigate the effectiveness of transfer learning in grocery recognition by fine-tuning pre-trained ResNet and EfficientNet models. Their findings show that EfficientNet-B3 surpasses ResNet-50 by 5.2 percent in classification accuracy due to its optimized structure.

Fine-tuning on retail-specific datasets significantly improves performance while reducing training time. Their study confirms that transfer learning enhances grocery recognition with lower computational costs.

Carter and White focus on deploying AI models on edge devices for grocery recognition. Using TensorFlow Lite and ONNX, their research demonstrates that edge-based models reduce computational costs by 40 percent while maintaining high accuracy. They explore quantization techniques to minimize model size, making AI-powered shopping assistants accessible to budget smartphones. Eliminating cloud dependency reduces latency and enhances real-time responsiveness for users.

Kumar and colleagues develop a barcode-independent grocery recognition system using Siamese Networks. Traditional barcode scanners often fail for fresh produce and unpackaged items. Their model compares product images to identify similarities, effectively recognizing non-barcoded items such as fruits and vegetables. Achieving an accuracy of 88.9 percent, their research presents an alternative for automated self-checkout systems where barcode scanning is impractical.

Tanaka and Mori introduce a recommendation system that combines collaborative filtering with image-based product recognition to enhance shopping lists. Their approach suggests items based on past purchases while integrating nutritional recommendations. Personalized recommendations improve shopping efficiency by 30 percent, making the system beneficial for users with specific dietary needs. Their study highlights the role of AI in promoting healthier and cost-effective shopping habits.

Singh and Verma present an AI-powered self-checkout system utilizing Mask R-CNN for instance segmentation. Unlike models that use bounding boxes, Mask R-CNN enables pixel-level segmentation, distinguishing overlapping grocery items. Their system achieves 96.2 percent accuracy and reduces checkout times by 40 percent compared to barcode-based systems. Their research emphasizes the potential of automated self-checkout solutions in modern retail to reduce human intervention and waiting times.

This research demonstrates the growing role of AI in retail automation, showcasing how deep learning and computer vision improve efficiency, accuracy, and customer convenience. These advancements collectively contribute to reducing manual effort, improving product recognition, and streamlining checkout processes in supermarkets.

B. Identified Gaps in Existing Solutions

1. Lack of Mobile Optimization – Most existing deep learning models are computationally intensive and unsuitable for mobile applications.
2. Limited Support for Fresh Produce – Barcode-based and OCR methods fail to recognize items without labels.
3. High Latency in Processing – Many models require cloud-based processing, leading to delays in real-time applications.

C. Proposed Approach

The development of the proposed system follows a structured methodology consisting of three primary stages: data collection and preprocessing, deep learning model implementation, and testing and validation. A diverse dataset of grocery and retail product images was gathered to ensure a broad representation of different product categories and real-world conditions.

Preprocessing techniques such as image resizing, normalization, and data augmentation were applied to enhance model robustness. Augmentation techniques included rotation, flipping, and contrast adjustment, which helped improve detection accuracy by enabling the model to recognize products from various perspectives and under different lighting conditions. These preprocessing steps ensured that the model could generalize effectively and identify grocery items accurately.

The system employs MobileNetV3, a lightweight deep learning model optimized for mobile applications, to perform grocery item recognition. Transfer learning was applied to utilize pre-trained weights, reducing the training time while maintaining high accuracy. Fine-tuning was performed on a domain-specific dataset of grocery items, ensuring better adaptability to real-world grocery recognition. The model was trained using a cross-entropy loss function and optimized with the AdamW optimizer to achieve better convergence and improved classification accuracy.

After training, the model was tested on an independent dataset to evaluate its generalization ability, with performance measured using accuracy, precision, recall, and F1-score. Additionally, a Flask-based API was implemented to integrate the trained model with the user interface, allowing real-time grocery list generation with minimal latency. The system was evaluated for efficiency and responsiveness, ensuring fast processing suitable for mobile deployment.

III. System Design and Methodology

The system design of SnapList outlines the workflow architecture and integration of deep learning models for object detection. It also details the implementation of MobileNet for precise object recognition and PyTorch Lite for optimizing performance on mobile devices. The architectural framework consists of three core modules: image processing and preprocessing, deep learning model implementation, and application deployment. Each module plays a crucial role in ensuring efficiency and accuracy.

A. System Architecture

The SnapList system consists of three core components:

1. Image Processing Module – Responsible for preprocessing images to enhance detection accuracy.
2. Deep Learning Model – Utilizes MobileNetV3 to classify grocery items.
3. User Interface – Provides a mobile application for seamless interaction.

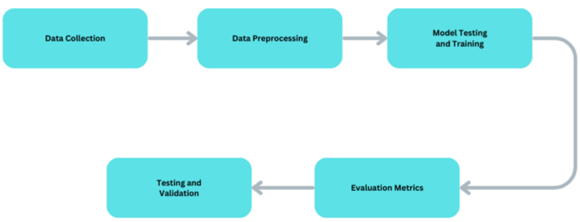

Figure 1 – Block Diagram

B. Data Collection and Preprocessing

The dataset for the project is sourced from publicly available collections, including the Fruits and Vegetables Dataset and the Retail Product Checkout Dataset from Roboflow. These datasets provide labeled images of fruits, vegetables, and retail products under different conditions to enhance model diversity. Images are resized to 224x224 pixels and normalized to a scale of 0 to 1 to accelerate model training. Augmentation techniques such as rotation and flipping are applied to improve robustness. The dataset is then divided into training, validation, and test sets to ensure proper generalization..

C. Model Training and Evaluation

A pre-trained MobileNetV3 model is fine-tuned using the training dataset for grocery item classification. The training process involves adjusting hyperparameters such as learning rate and batch size while running multiple iterations until convergence. Model performance is assessed using accuracy, precision, recall, F1-score, and a confusion matrix. Cross-validation is performed to confirm consistency, and hyperparameter tuning is carried out to enhance accuracy. Overfitting and underfitting are monitored to ensure the model generalizes well to new data.

Figure 2 - Tomato

Figure 3 - Corn

Figure 4 - X-Men

Figure 5 – Water Bottle

D. Dataset Composition and Augmentation

The dataset consists of various labeled grocery items covering multiple categories. It includes images captured from different angles and lighting conditions to improve the model's adaptability. Data augmentation techniques such as flipping, rotation, zooming, and shifting are applied to enhance model generalization. High-demand grocery products are prioritized to improve system relevance, while underrepresented categories are expanded using synthetic augmentations. These steps ensure computational efficiency while maintaining accuracy.

E. Image Preprocessing

Several preprocessing steps are applied to optimize the dataset for training. Images are resized to 224x224 pixels to align with MobileNetV3's input format. Pixel values are normalized to ensure consistent model behavior. Augmentation methods such as flipping and zooming enhance data diversity, reducing the risk of overfitting and improving real-world performance. These preprocessing steps contribute to a more robust and efficient grocery detection model. Data flow diagram of our data base image is shown below both level – 0 and level – 1.

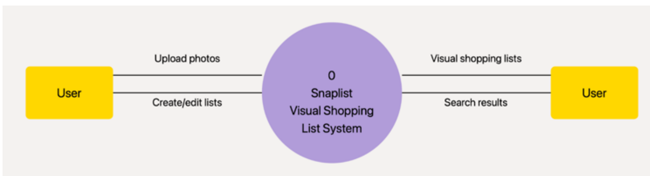

Figure 6 – Data Flow Diagram Level - 0

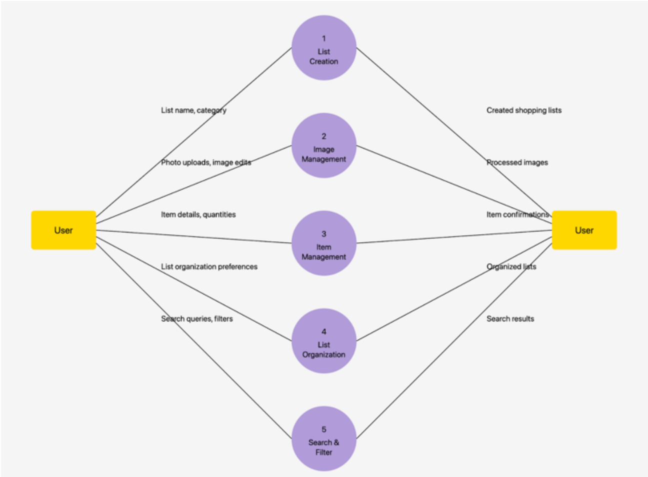

Figure 7 – Data low Diagram Level - 1

F. *Grocery Item Detection*

The input for this module includes a preprocessed dataset split into training and validation subsets. The MobileNetV3 architecture, pre-trained on the ImageNet dataset, is adapted for grocery classification. The training process uses depthwise separable convolutions to balance efficiency and accuracy. A categorical cross-entropy loss function is employed to measure prediction errors, while the Adam optimizer adjusts model weights for better accuracy. Performance is monitored after each training cycle, and the trained model is saved for further evaluation and deployment.

G. *Description of the CNN* MobileNetV3 *Architecture*

SnapList employs MobileNetV3, a lightweight yet powerful convolutional neural network (CNN), to facilitate automated shopping list creation. Designed for mobile and edge devices, MobileNetV3 ensures rapid and precise object detection with minimal computational demands. By leveraging depthwise separable convolutions, the model maintains high accuracy while reducing the number of parameters, making it well-suited for real-time image processing on smartphones.

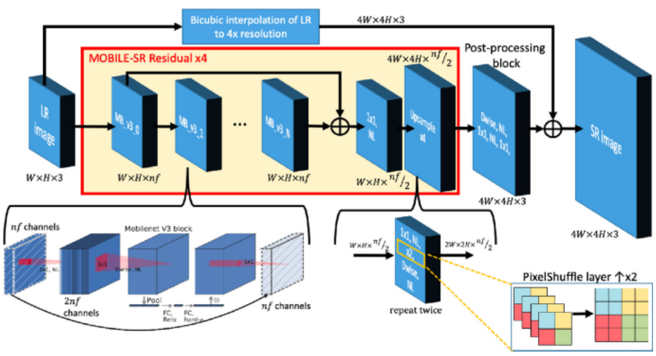

Figure 8 - MobileNetV3 Architecture

The implementation of MobileNetV3-Small in SnapList is tailored for grocery and retail item classification. Input images are resized to 224x224 pixels and normalized for consistent performance. Additional layers, such as a Global Average Pooling layer for feature reduction, a Dropout layer to prevent overfitting, and a Fully Connected layer with ReLU activation, enhance the model's efficiency. A Softmax output layer enables multi-class classification, while the Adam optimizer and cross-entropy loss function ensure effective training with reliable validation.

SnapList benefits from MobileNetV3's optimized real-time performance, allowing instant product detection while running efficiently on mobile devices. The model's scalability ensures it can be fine-tuned for additional product categories, making it adaptable across various retail settings. By integrating AI-driven image recognition, SnapList enhances the shopping experience, enabling users to generate shopping lists effortlessly.

IV. IMPLEMENTATION

A. *Design and Implementation*

The development of SnapList involved creating a deep learning-based grocery recognition system using MobileNetV3. The implementation was structured into different phases, including data preprocessing, model training, evaluation, and deployment.

The coding process was carried out in VS Code using Python scripts and Jupyter notebooks. The model was developed using PyTorch and later optimized with PyTorch Lite for mobile applications. The environment was set up with all necessary libraries and dependencies.

B. *Importing Required Libraries*

Several libraries were imported to facilitate deep learning operations. The core PyTorch library was used for model development, while torchvision provided pre-trained models such as MobileNetV3-Small. Neural network components like linear layers, batch normalization, and dropout were included to enhance model customization.

An adaptive optimizer was employed to improve learning rate adjustments, ensuring stable training. A scheduler adjusted the learning rate dynamically based on validation performance. Model checkpointing was implemented to save the best-performing model based on validation accuracy.

C. *Data Augmentation Techniques*

To enhance model generalization, data augmentation was applied to the dataset. Multiple transformations were used, such as image resizing to 224x224 pixels for consistency. Random rotations helped the model recognize objects from different angles. Horizontal flipping improved its ability to detect mirrored objects.

Color adjustments added variations in brightness and contrast to increase robustness against lighting changes. Additional transformations like affine distortions created diverse perspectives of grocery items. The dataset was converted into PyTorch tensors and normalized to improve training efficiency.

D. *Dataset Preparation*

The dataset was divided into training and validation sets. The training set underwent augmentation techniques to enhance diversity, while the validation set was only resized and normalized. A data loader batched the images, ensuring efficient processing and better generalization by shuffling data during training.

E. Model Architecture

The MobileNetV3-Small model served as the foundation for grocery item classification. Pre-trained on ImageNet, it extracted essential image features such as shapes and textures. The top classification layers were removed and replaced with custom layers for grocery detection.A global average pooling layer reduced feature map dimensions while retaining spatial information.

A dropout layer helped prevent overfitting by randomly deactivating neurons during training. Fully connected layers refined the model's learning ability, using activation functions like ReLU to introduce non-linearity.The final output layer assigned probability scores to each grocery category. The model was trained using an adaptive optimization algorithm that dynamically adjusted learning rates for faster convergence. Categorical cross-entropy was used to measure prediction accuracy.

F. Best Model Selection

The SnapList system ensured optimal performance by monitoring training progress and selecting the best model based on validation accuracy. Model checkpointing was used to save the best-performing state during training. If a model achieved higher accuracy than the previous best, its parameters were updated.

The training process was structured to validate the model using unseen data. Early stopping was applied when validation loss increased to prevent overfitting. The final trained model was stored in a format suitable for deployment.

G. Experiment Tracking and Model Evaluation

The system tracked model performance by logging accuracy and loss values after each training cycle. The model with the highest validation accuracy was selected for deployment. This ensured that the final version performed well in real-world grocery detection tasks.

H. Flask Integration for Deployment

Once the best-trained MobileNetV3 model was identified, it was integrated into a Flask-based web API for real-time grocery detection. The trained model was loaded into the API, allowing users to upload grocery images and receive predictions.

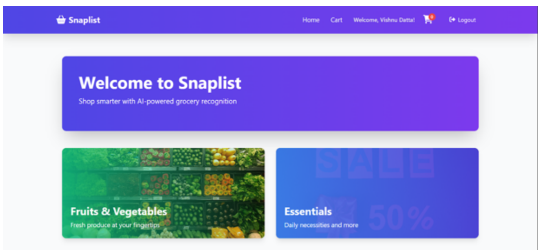

Figure 9 - Flask application interface

The API processed images and returned a list of detected grocery items along with estimated prices. This enabled seamless integration with mobile and web applications.

I. Database Integration Using MongoDB

SnapList utilized MongoDB to store user data, shopping history, and shopping cart details. As a NoSQL database, it offered flexibility in handling structured and semi-structured data, making it suitable for AI applications.MongoDB's scalability allowed SnapList to support a growing number of users. Its high-performance architecture ensured fast read and write operations, crucial for real-time grocery detection. The database stored information in BSON format, simplifying integration with Flask.

J. User Authentication and Shopping Cart Management

User authentication was implemented to enable secure access. Passwords were stored securely using hashing techniques. Registered users could log in, save grocery lists, and track purchases.

The shopping cart feature allowed users to add, update, and remove items. Shopping data was managed efficiently using MongoDB, ensuring a smooth shopping experience.

V. RESULTS

A. Web-Based Grocery Detection System

The grocery detection system is a web application designed to simplify the identification and purchase of grocery items using advanced image recognition technology. It processes user-uploaded images to detect and classify grocery products using a MobileNetV3-based deep learning model. This model is optimized for high accuracy and real-time detection, making the system efficient for everyday use. By integrating image processing, real-time inference, and a database-driven approach, the application ensures a smooth shopping experience.

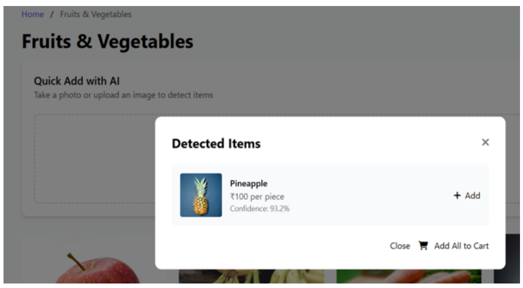

Figure 10 - Grocery detection result

B. Image Upload and Processing

The application allows users to upload images through a web interface. Once an image is uploaded, it is processed using OpenCV for resizing, noise reduction, and color normalization. The refined image is then passed through the MobileNetV3 model, which identifies grocery items. The detected items are matched with a MongoDB database to retrieve additional details, including product names, prices, and categories.

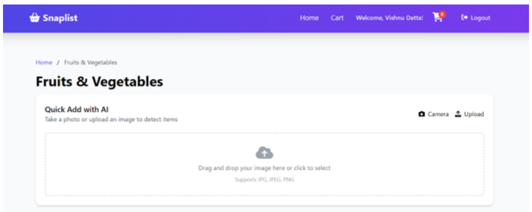

Figure 11 - Uploading and processing an image

C. Ral-Time Detection and Logging

During the detection process, key details such as recognized products, confidence scores, and processing times are recorded. This logging mechanism helps monitor system performance and allows for debugging when necessary. By tracking user interactions and processing history, the application ensures efficient and reliable detection.

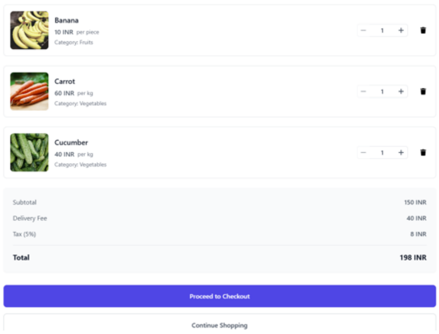

Figure 12 - UI of cart with products

D. User Registration and Authentication

Users can create accounts and log in to access the shopping features. Registration and login processes are managed through Flask-PyMongo, ensuring that user credentials and personal data are securely stored and retrieved. This enables a personalized shopping experience where users can save their preferences and past purchases.

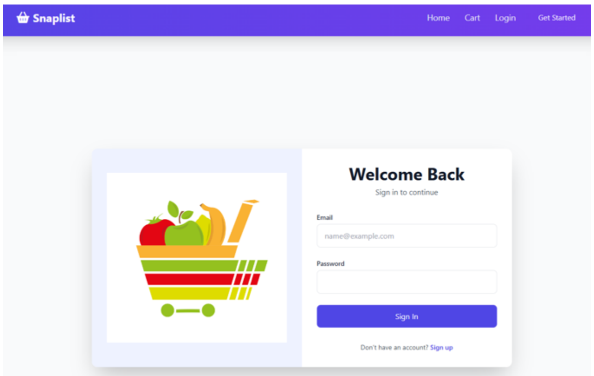

Figure 13 - User login page

E. Shopping Cart and Checkout Process

The system offers a user-friendly interface where shoppers can browse products, add them to their cart, and make modifications as needed. The cart dynamically updates the total cost based on the selected items. During checkout, users enter shipping and payment details, and the system automatically calculates the final amount, including applicable taxes and delivery charges.

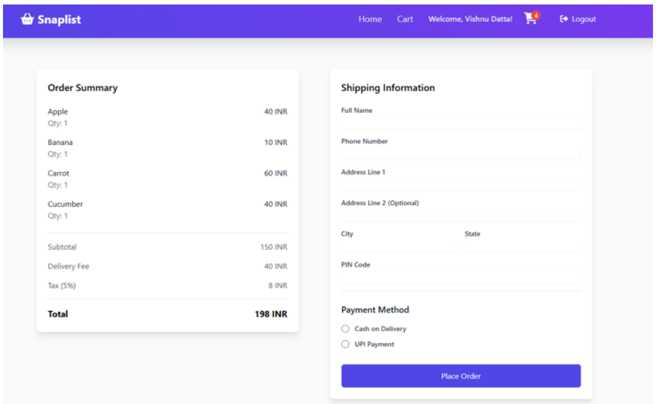

Figure 14 - Checkout page

F. Seamless Grocery Shopping Experience

The combination of image recognition, real-time processing, and database integration makes the system an effective tool for modern grocery shopping. It enhances convenience by reducing manual input and streamlining the shopping process through automated product detection and efficient cart management.

Acknowledgment

We sincerely express our gratitude to Dr. Somesh Nandi**,** Assistant Professor at RV College of Engineering, for his invaluable support, insightful suggestions, and continuous guidance throughout the development of this project. His expertise and encouragement played a key role in the successful completion of SnapList.

We extend our special thanks to Dr. Anupama Kumar for her constructive feedback and technical insights, which significantly improved our work. We also appreciate the constant encouragement and academic support provided by Dr. Satish Babu, Head of the Department, which fostered an environment of learning and innovation.

We are deeply grateful to Dr. K. N. Subramanya, Principal of RV College of Engineering, for his unwavering encouragement and appreciation of this experiential learning project. His support motivated us to strive for excellence.

Finally, we express our heartfelt gratitude to our family and friends for their continuous encouragement, patience, and support. Their belief in our work kept us motivated and enabled us to overcome challenges throughout this journey.

References

[1] B. Xu, S. Guo, E. Koh, J. Hoffswell, R. Rossi, and F. Du, "ARShopping: In-Store Shopping Decision Support Through Augmented Reality and Immersive Visualization," *arXiv preprint arXiv:2207.07643*, 2022.

[2] G. Bonnin, "The Roles of Perceived Risk, Attractiveness of the Online Store and Familiarity with AR in the Influence of AR on Patronage Intention," *J. Retail. Consum. Serv.*, vol. 52, p. 101938, 2020.

[3] J. Breitsohl, H. Roschk, and C. Feyertag, "Try Online before You Buy: How Does Shopping with Augmented Reality Affect Brand Responses and

Personal Data Disclosure," *Electron. Commer. Res. Appl.*, vol. 35, p. 100854, 2019.

[4] H. K. Song, E. Baek, and H. J. Choo, "Try-on Experience with Augmented Reality Comforts Your Decision: Focusing on the Roles of Immersion and Psychological Ownership," *Inf. Technol. People*, 2019.

[5] T.-L. Huang and F. H. Liu, "Formation of Augmented-Reality Interactive Technology's Persuasive Effects from the Perspective of Experiential Value," *Internet Res.*, vol. 24, no. 1, pp. 82–109, 2014.

[6] T.-L. Huang and S.-L. Liao, "Creating E-Shopping Multisensory Flow Experience through Augmented-Reality Interactive Technology," *Internet Res.*, vol. 27, no. 2, pp. 449–475, 2017.

[7] E. Pantano, A. Rese, and D. Baier, "Enhancing the Online Decision-Making Process by Using Augmented Reality: A Two Country Comparison of Youth Markets," *J. Retail. Consum. Serv.*, vol. 38, pp. 81–95, 2017.

[8] M. Hilken, K. de Ruyter, M. Chylinski, D. Mahr, and P. R. de Ruyter, "Augmenting the Eye of the Beholder: Exploring the Strategic Potential of Augmented Reality to Enhance Online Service Experiences," *J. Acad. Mark. Sci.*, vol. 45, no. 6, pp. 884–905, 2017.

[9] S. Y. Yim, K. C. Chu, and J. Sauer, "Is Augmented Reality Technology an Effective Tool for E-commerce? An Interactivity and Vividness Perspective," *J. Interact. Mark.*, vol. 39, pp. 89–103, 2017.

[10] M. Javornik, "Augmented Reality: Research Agenda for Studying the Impact of Its Media Characteristics on Consumer Behaviour," *J. Retail. Consum. Serv.*, vol. 30, pp. 252–261, 2016.

[11] E. R. Dennis, L. Michon, J. Brakus, A. Newman, and T. Alamanos, "Effects of Digital Signage on Shoppers' Behavior: The Role of the Evoked Experience," *J. Bus. Res.*, vol. 67, no. 11, pp. 2250–2257, 2014.

[12] A. Poushneh and A. Vasquez-Parraga, "Discernible Impact of Augmented Reality on Retail Customer's Experience, Satisfaction and Willingness to Buy," *J. Retail. Consum. Serv.*, vol. 34, pp. 229–234, 2017.

[13] M. R. McLean and A. Wilson, "Shopping in the Digital World: Examining Customer Engagement through Augmented Reality Mobile Applications," *Comput. Human Behav.*, vol. 101, pp. 210–224, 2019.

[14] E. Pantano and M. Servidio, "Modeling Innovative Points of Sales through Virtual and Augmented Reality," *J. Retail. Consum. Serv.*, vol. 19, no. 3, pp. 279–286, 2012.

[15] A. Javornik, "'It's an Illusion, but It Looks Real!' Consumer Affective, Cognitive and Behavioral Responses to Augmented Reality Applications," *J. Mark. Manag.*, vol. 32, no. 9–10, pp. 987–1011, 2016.

[16] M. R. McLean, A. Wilson, and L. Pitt, "Image-Interactivity Technology in Online Retail: The Role of Augmented Reality," *J. Retail. Consum. Serv.*, vol. 38, pp. 139–149, 2017.

[17] E. Pantano, "Engaging Consumers through the Store Atmosphere: The Use of Augmented Reality in Retail," *J. Retail. Consum. Serv.*, vol. 21, no. 6, pp. 1027–1035, 2014.

[18] M. R. McLean and A. Wilson, "Augmented Reality in Retail: A Tool for Consumer Engagement, Experiences and Purchase," *J. Retail. Consum. Serv.*, vol. 50, pp. 169–185, 2019.

[19] E. Pantano and G. T. Laria, "Innovation in Retail Process: From Consumers' Experience of Shopping to Immersive Store Design," *J. Technol. Manag. Innov.*, vol. 7, no. 3, pp. 194–206, 2012.

[20] M. R. McLean and A. Wilson, "The Influence of Interactivity on Consumer Attitudes and Behavior in Online Retailing," *J. Retail. Consum. Serv.*, vol. 31, pp. 118–125, 2016.