



**RV College of  
Engineering®**

***Go, change the world***

## **Department of AIML**

### **SnapList: Visual Shopping List Creator**

**Presented by**

**Students Name and USN**

<b>Kota Vishnu Datta</b>	<b>- 1RV22AI024</b>
<b>Chillale Naveen</b>	<b>- 1RV22AI013</b>
<b>D Sai Siva Bhaswanth</b>	<b>- 1RV22AI016</b>

**Faculty Mentors: Prof. Somesh Nandi  
Dr S Anupama Kumar**



# Agenda

1. Agenda
2. Introduction
3. Literature Survey
4. Summary of LS
5. Requirement analysis – hardware and software specification
6. System architecture ( -- ANN-DL architecture ) Eg .. architecture of CNN
7. Methodology
8. Module specification –
  - a. Module 1 : data collection and pre processing
    - i. Input ii. Process iii output
  - b. Module 2 : Implementation of ANN / DL algorithm
    - i. Input ii. Process iii output
  - c. Module 3 : testing and validation
    - i. Input ii. Process iii output

# Introduction

*Go, change the world*

- The rapid advancements in Deep learning and Computer Vision have enabled groundbreaking applications in retail and consumer services.
- These technologies have significantly transformed how users interact with digital systems, offering enhanced automation and efficiency.
- Traditional shopping list applications rely heavily on manual input, leading to inefficiencies.
- SnapList: Automates shopping list creation via image recognition for a seamless, personalized experience.
- Leverages MobileVNet for accurate detection for mobile optimization.



# Introduction

Go, change the world

- SnapList aligns with the global shift towards automation, addressing the growing need for AI-powered solutions in daily life
- Studies show that 85% of retail businesses plan to adopt AI solutions by 2025 to enhance customer experience and operational efficiency (source: Gartner).
- A survey revealed that 67% of shoppers prefer applications that minimize manual effort, highlighting the relevance of automated tools like SnapList (source: Statista).
- Personalization in retail boosts customer satisfaction and loyalty, with 80% of consumers more likely to purchase from brands offering personalized experiences (source: McKinsey).





# Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
1.	<p>Smart Shopping and Cart Billing System Using Deep Learning</p> <ul style="list-style-type: none"><li>Authors: Ramkumar S, Saravanan R, Kanagaraj Venusamy, Rani Jabbar, N. Jeevitha</li></ul>	<ul style="list-style-type: none"><li>Conference: Proceedings of the Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI 2024)</li><li>Publisher: IEEE</li></ul>	<p>This paper proposes a smart shopping cart that automates billing using computer vision and deep learning to reduce checkout queues. Equipped with a Raspberry Pi, webcam, LCD, and Arduino-controlled motors, the cart autonomously follows users via face detection. Products are scanned in real-time, and payments can be made via QR codes, UPI, or cards, with bills sent via email/SMS. This system improves efficiency and accuracy over traditional barcode-based methods by minimizing manual intervention.</p>
2.	<p>Batch Normalization Free Rigorous Feature Flow Neural Network for Grocery Product Recognition</p> <ul style="list-style-type: none"><li>Authors: Prabu Selvam, Muhammad Faheem, Vidyabharathi Dakshinamurthi, Akshaj Nevgi, R. Bhuvaneswari, K. Deepak, Joseph Abraham Sundar</li></ul>	<ul style="list-style-type: none"><li>Journal: IEEE Access</li><li>Volume: 12, 2024</li><li>Publisher: IEEE</li></ul>	<p>This paper presents BNFRNN, a batch normalization-free neural network for grocery product recognition. It uses a three-stage pipeline: YOLOv5 for detection, an OD-Refiner to correct bounding box errors, and an OCR-based recognizer for extracting product details. Tested on the WebMarket dataset, it achieved 92.56% precision, 85.64% recall, and 88.97% F-score, outperforming existing methods while improving computational efficiency.</p>





# Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
3.	Y. Wu, D. Li, "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution"	IEEE, 2021	DetectoRS introduces two novel components: recursive feature pyramids and switchable atrous convolutions. These methods enhance multi-scale feature extraction and improve detection accuracy across a variety of objects and scenes. The architecture achieves state-of-the-art object detection performance.
4.	X. Zhang, W. Lin, "Revisiting ResNets: Improved Training and Architectural Refinements"	ICCV, 2021	This paper revisits the ResNet architecture and introduces new training techniques and minor architectural changes to enhance performance. The authors achieve state-of-the-art results in various computer vision tasks by improving both model efficiency and accuracy.
5.	A. Dosovitskiy, M. Schmidt, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"	IEEE,2021	The paper presents Vision Transformers (ViT), a novel architecture that replaces convolutions with transformers for image classification tasks. The authors demonstrate that ViTs outperform CNNs on large datasets, proving the potential of transformers for vision tasks.



# Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
6.	Y. LeCun, S. K. Nair, "Efficient Neural Network Architectures for On-Device Vision Applications"	CVPR, 2021	Focuses on developing efficient neural network architectures for deployment on mobile and embedded devices. The paper demonstrates techniques for reducing model size and computational load while maintaining high accuracy in on-device applications.
7.	K. Han, Z. Yang, "GhostNet: More Features from Cheap Operations"	IEEE, 2020	GhostNet is a lightweight architecture designed for mobile and embedded devices. It uses cheap operations like ghost modules to efficiently capture features without a significant increase in computation. The paper shows how this network can achieve high performance with fewer resources.
8.	Z. Zhang, H. Liu, "ResNeSt: Split-Attention Networks"	CVPR, 2020	ResNeSt introduces split-attention blocks into the ResNet architecture, improving feature representation for a wide range of vision tasks. The authors show that this modification leads to significant improvements in accuracy and generalization ability across various benchmarks.



# Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
9.	H. Liu, L. Zheng, "Weakly Supervised Object Detection Using Deep Feature Aggregation"	IEEE, 2019	This paper presents a weakly-supervised object detection framework that aggregates deep features to improve detection accuracy without full supervision. The method is shown to perform competitively with fully supervised methods while requiring fewer labeled samples.
10.	M. Tan, Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"	ICML, 2019	The authors propose EfficientNet, a model that optimizes the scaling of CNN depth, width, and resolution. EfficientNet achieves state-of-the-art performance while being computationally more efficient than previous architectures like ResNet and Inception.
11.	T.-U. İk, H. H. Olayiwola, "Smart and Accurate Shopping Cart System"	MOE Taiwan, 2018	The paper presents an enhanced smart shopping cart system using YOLOv2 and Faster R-CNN for object detection. The system incorporates smoothing techniques to improve detection accuracy, enabling efficient shopping experience by automatically identifying items in the cart.





# Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
12.	J. Yu, L. Xie, "BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation"	IEEE, 2018	BiSeNet proposes a network that separates the spatial and context paths to achieve real-time, high-quality semantic segmentation. The method achieves efficiency by focusing on important regions in the input image while maintaining accurate predictions across the entire image.
13.	X. Huang, H. L. Zhang, "Multimodal Unsupervised Image-to-Image Translation"	ECCV, 2018	The authors present a GAN-based framework for unsupervised image-to-image translation, handling multimodal outputs like generating diverse images from a single input. The method helps improve the diversity and realism of translations in applications such as image restoration and style transfer.
14.	B. Li, W. Y. Zhang, "CSRNet: Dilated Convolutional Networks for Crowd Counting"	CVPR, 2018	The paper introduces CSRNet, a novel method for accurate crowd density estimation using dilated convolutional networks. The approach effectively captures multi-scale contextual information, resulting in a model that outperforms conventional methods on crowd counting benchmarks.



# Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
15.	J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement"	arXiv preprint, 2018	YOLOv3 significantly improves object detection speed and accuracy compared to previous versions. The paper discusses improvements to the architecture, incorporating multi-scale predictions and better network layers to achieve real-time detection with enhanced precision in complex scenes.
16.	L. Zheng, Z. Yu, "Person Re-Identification in the Wild"	CVPR, 2017	This work focuses on the challenges of person re-identification (Re-ID) in real-world scenarios, such as varying poses and occlusions. The authors propose a new dataset for benchmarking and provide baseline methods that improve Re-ID accuracy across different conditions.
17.	T.-Y. Lin, M. G. Lee, "Focal Loss for Dense Object Detection"	ICCV, 2017	The authors introduce focal loss, a loss function that addresses class imbalance in dense object detection tasks. By focusing more on hard-to-detect objects, the method significantly improves the performance of dense detectors in complex environments with various object scales.



# Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
18.	M. He, A. G. B. Kwan, "Deep Feature Flow for Video Recognition"	CVPR, 2017	The paper introduces the Deep Feature Flow (DFF) framework, which significantly accelerates video recognition by reusing visual features across frames. The method helps improve the speed and accuracy of video processing models in real-time applications, such as surveillance and sports analytics.
19.	B. Zhou, A. Torralba, "Learning Deep Features for Discriminative Localization"	CVPR, 2016	Introduces Class Activation Mapping (CAM), which utilizes CNN-based heatmaps for weakly-supervised object detection. CAM provides a deeper understanding of where CNNs focus their attention, thus enhancing interpretability and improving detection accuracy in visually complex datasets.
20.	Z. Wang, A. Q. Zhang, "Perceptual Losses for Real-Time Style Transfer"	ECCV, 2016	The authors propose using perceptual loss functions within CNNs to improve style transfer tasks in real-time, enhancing image quality while reducing computation time. The method also contributes to image super-resolution tasks by optimizing perceptual similarity rather than pixel-wise errors.



## Similarities

- **Focus on Deep Learning Models:** Most of the papers utilize deep learning techniques such as CNNs, RNNs, GANs, and Transformers for tasks like image recognition, object detection, and segmentation.
- **Benchmark Datasets:** Many papers (e.g., "YOLOv3," "ResNeSt," "EfficientNet") use standard datasets like ImageNet, Cityscapes, and COCO for evaluation.
- **Real-Time Applications:** Several studies, such as "BiSeNet" and "CSRNet," emphasize real-time capabilities for tasks like crowd counting and segmentation.

## Differences

- **Architectural Innovations:** While YOLOv3 and DetectoRS focus on improving object detection accuracy and speed, papers like "EfficientNet" and "GhostNet" prioritize resource efficiency for deployment on mobile devices.
- **Supervision Levels:** Some papers, such as "Weakly Supervised Object Detection," explore methods to reduce dependency on fully labeled datasets, while others like "ResNet" rely on fully supervised learning.



## Accuracy Improvements

- High-Accuracy Architectures: Papers like "YOLOv3," "DetectoRS," and "ResNeSt" report state-of-the-art accuracy by introducing innovative modules like recursive feature pyramids and split-attention blocks.
- Efficiency vs. Accuracy Trade-Off: "EfficientNet" and "GhostNet" focus on balancing accuracy and computational efficiency, often achieving comparable performance with fewer parameters than traditional models.



# Objectives

Go, change the world

- Develop an AI-powered shopping list application that recognizes and categorizes products from images.
- Enhance user experience by providing a seamless visual way to create and manage shopping lists.
- Reduce manual input by using computer vision to detect and list products automatically.
- Improve shopping efficiency by allowing users to organize, share, and check off items in real time.





## Hardware Specifications

### 1. CPU:

- Minimum: Intel i5 or AMD Ryzen 5 (Quad-core)
- Recommended: Intel i7/i9 or AMD Ryzen 7/9 for faster data preprocessing.

### 2. GPU:

- Minimum: NVIDIA GPU with CUDA support, such as GTX 1060 (6GB VRAM).
- Recommended: NVIDIA RTX 3060 or higher (12GB VRAM or more) to handle deep learning tasks efficiently.

### 3. RAM:

- Minimum: 8GB.
- Recommended: 16GB or more for handling large datasets and parallel processing.

### 4. Storage:

- Minimum: 256GB SSD for software and dataset storage.
- Recommended: 1TB SSD or HDD for large-scale image datasets.

### 5. Software Dependencies:

- Operating System: Ubuntu 20.04 LTS or Windows 10 with WSL.
- Frameworks: Caffe (with VisNet modifications), CUDA Toolkit (compatible with GPU), and Python for auxiliary scripts.



# Requirement analysis

## Software Specifications

### Operating Systems

- Minimum: Ubuntu 18.04 LTS or Windows 10 (64-bit)
- Recommended: Ubuntu 20.04 LTS or higher for better compatibility with machine learning frameworks.

### Programming Languages

- Primary: Python 3.7+ for auxiliary scripts and integration.
- Secondary: C++ for CUDA-based modules like K-Nearest Neighbor (KNN) search.

### Frameworks and Libraries

- **Deep Learning Framework:**
  - Caffe: Required for training the VisNet model with modifications for image augmentation and triplet accuracy layers
  - TensorFlow/PyTorch (Optional): For testing alternative approaches or additional features.
- **CUDA Toolkit:**
  - CUDA 10.2 or higher for GPU acceleration of KNN and deep learning tasks.



# Requirement analysis

## Software Specifications

### Frameworks and Libraries

#### Python Libraries:

- NumPy (1.21.6 or later): For numerical computations, including array manipulation and mathematical operations.
- OpenCV (4.5.3 or later): For image preprocessing, transformations, and feature extraction tasks.
- Matplotlib (3.4.3 or later)/Seaborn (0.11.2 or later): For creating plots, charts, and advanced data visualizations.
- Scikit-learn (1.0.2 or later): For auxiliary machine learning tasks, including clustering, evaluation metrics, and preprocessing.
- Pandas (1.3.3 or later): For managing and processing structured data, such as CSV files or DataFrames.

#### Data Processing:

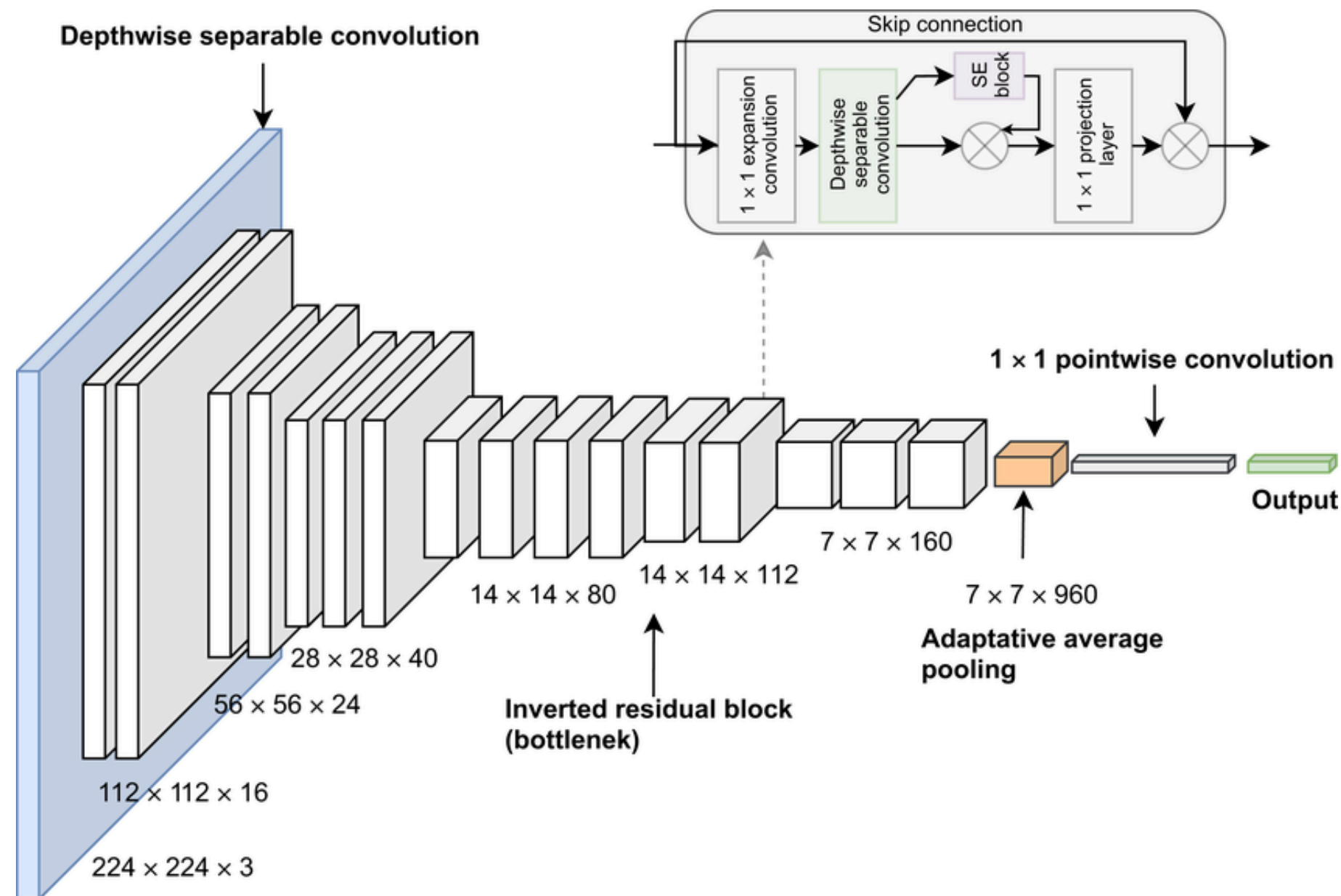
- Triplet Sampling Code: Prepares datasets for the training pipeline, requiring compatible file structures.

#### Visualization Tools:

- Tools like TensorBoard (optional) for monitoring training progress.

## MobileNetV3 Architecture

The MobileNetV3 architecture, a lightweight and efficient convolutional neural network (CNN), is utilized to implement the deep learning solution for automated shopping list creation in SnapList. MobileNetV3 is optimized for mobile and edge devices, ensuring fast and accurate object detection while maintaining minimal computational overhead.







## MobileNetV3 Architecture

### 1. Input Layer

- Accepts image inputs, such as photos of shopping items.
- The network takes input images resized to 224x224 pixels with three color channels (RGB). Images are normalized for consistent performance across different lighting conditions.

### 2. Base Model:

- The MobileNetV3-Small architecture is used as the base model, with modifications for retail and grocery item classification.

### 3. Additional layers include:

After the base model, additional layers were added, including:

- Global Average Pooling Layer: Reduces feature map dimensions while retaining spatial information.
- Dropout Layer: Introduced to prevent overfitting by randomly disabling certain neurons during training.
- Fully Connected Layer: Contains 512 neurons (ReLU activation) for feature extraction.
- Softmax Output Layer: Classifies images into their respective retail or grocery categories.
- Results are displayed as a list or gallery, enabling users to add matched items to their shopping list.



## MobileNetV3 Architecture

### Benefits of Using MobileNetV3 in SnapList :

- **Real-Time Performance:** Optimized for low-latency inference, enabling instant grocery and product detection.
- **Mobile-Friendly:** Designed to run efficiently on mobile and edge devices without excessive power consumption.
- **Scalability:** Can be fine-tuned for additional products, making it adaptable to various retail environments.



## 1. Data Collection

- The data for the project is sourced from publicly available datasets: **Fruits and Vegetables Dataset**, **Retail Product Checkout Dataset** from Roboflow. These datasets contain images of fruits, vegetables, and retail products labeled with their corresponding categories, ensuring diversity in object types and environmental conditions.

## 2. Data Preprocessing

- **Resizing:** All images were resized to 224x224 pixels to align with MobileNetV3's input format.
- **Normalization:** Pixel values were normalized for consistent model behavior.
- **Augmentation:** Techniques such as rotation, flipping, zooming, and shifting were used to improve model generalization.



## 3. Model Training

- **Architecture:**

- Use the MobileNetV3 model, a lightweight CNN optimized for mobile applications, with modifications such as:
  - Depth-wise separable convolutions for computational efficiency.
  - Batch normalization layers to stabilize training.
- Train using a supervised classification approach, optimizing for multi-class detection.

- **Loss Function:** Optimize with categorical cross-entropy loss, ensuring accurate class predictions across various grocery items.

- **Framework:** Implement using PyTorch, with additional optimizations such as quantization-aware training for mobile deployment.

- **Hardware:** Utilize GPUs with CUDA acceleration for faster training and inference, enabling real-time grocery item recognition.

## 4. Feature Embedding

- Extract numerical representations (embeddings) of grocery item images through the trained MobileNetV3 model.
- Store embeddings in a vector database for fast retrieval and similarity-based recommendations during inference.



## 5. System Integration

### Integration Process:

- Combine the trained model with a user interface or Flask API.

### Input Handling:

- Accept user-provided images.
- Preprocess and forward them to the model for feature extraction and matching.

### Output:

- Display ranked results (visually similar items) for user selection.

## 6. Evaluation and Validation

- Metrics:
  - Classification Accuracy: Measures the model's ability to correctly classify grocery items.
  - Mean Average Precision (mAP): Evaluates ranking quality for retrieval.
- Testing Dataset: Validate on unseen data to measure generalization.



# Module Specification

## Module 1: Data Collection and Preprocessing

### Input: Raw Image Data

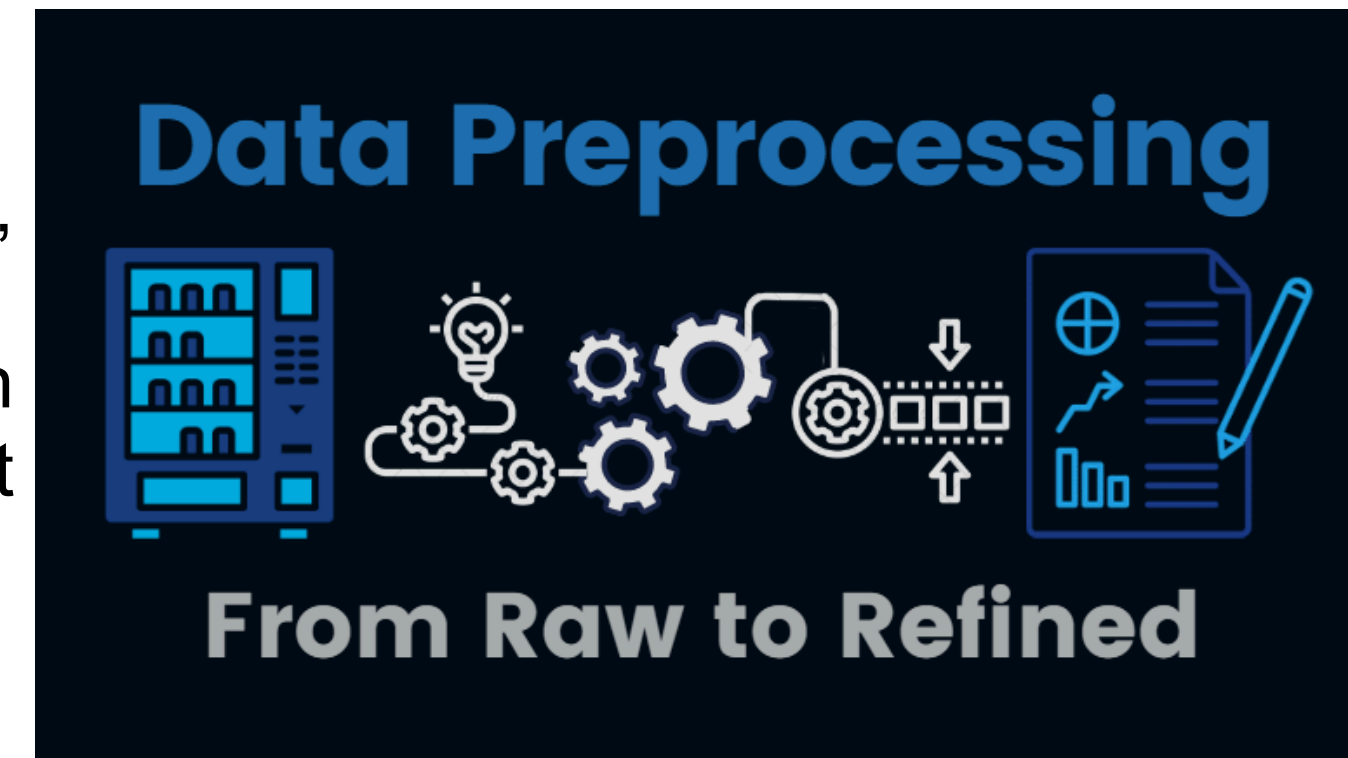
- The inputs for this module consist of images of grocery items, including fruits, vegetables, and retail products.
- These images are sourced from publicly available datasets such as the Fruits and Vegetables Dataset and the Retail Product Checkout Dataset from Roboflow.

### Process: Image Preprocessing

- **Resizing:** Images are resized to a standardized dimension of 224x224 pixels to ensure compatibility with the MobileNetV3 model.
- **Normalization:** Pixel values are scaled to a range of 0 to 1, enhancing consistency across the dataset and optimizing performance during training.
- **Data Augmentation:** Techniques such as flipping, rotation, and zooming are applied to artificially expand the dataset.

### Output: Preprocessed Image

- The output of this module is a preprocessed dataset that is clean, normalized, and ready to be used for training and testing.





## Module 2: Implementation of ANN / DL Algorithm

### Input:

- The preprocessed dataset, divided into training and validation subsets.
- The network configuration, which in this case is the MobileNetV3 architecture.

### Process: Deep Learning Algorithm Application

- **Model Construction:** The MobileNetV3 architecture is adapted for multi-class classification of grocery items. Depthwise separable convolutions are leveraged to optimize efficiency and accuracy.
- **Training:** The model is trained using the preprocessed dataset. A categorical cross-entropy loss function is employed to measure prediction errors, while the Adam optimizer adjusts model weights for improved accuracy.
- **Validation:** After each training epoch, the model's performance is evaluated on the validation subset to monitor progress and detect potential overfitting.
- **Model Saving:** The trained model is saved in a format such as .pt, enabling further testing, evaluation, and eventual deployment.



# Module Specification

## Module 2: Implementation of ANN / DL Algorithm

### Output:

The output of this module is a trained MobileNetV3 model file containing the learned weights and architecture. Additionally, training logs, including loss and accuracy curves, are generated to track performance across epochs.

# Module Specification

Go, change the world

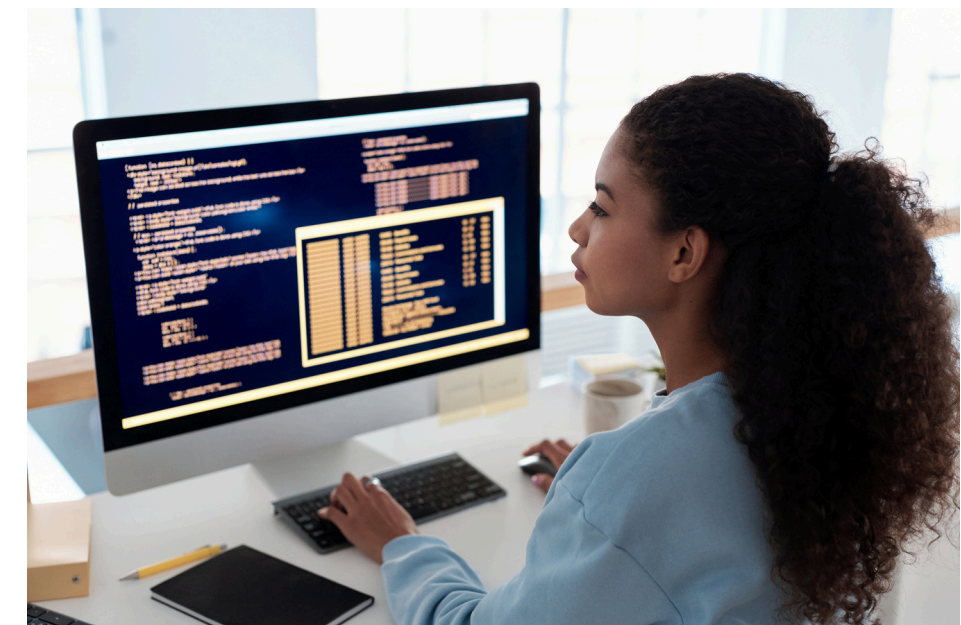
## Module 3: Testing and Validation

### Input: Model Predictions or Features

- The outputs from the trained model are evaluated using test data that was not used during training.

### Process: Testing and Validation

- **Testing:** The trained model is tested on unseen images from the test dataset. Predictions generated by the model are compared with the actual ground truth labels.
- **Validation:** The model's generalizability is evaluated by analyzing performance on diverse test samples. Misclassifications are closely examined to identify potential weaknesses.
- **Performance Metrics:** Use accuracy, precision, recall, F1 score, or confusion matrices to evaluate classification performance.
- **Cross-validation:** To assess the model's generalization ability across different datasets.



### Output: Validation Results

- A detailed report showing the model's accuracy, precision, and other relevant metrics.

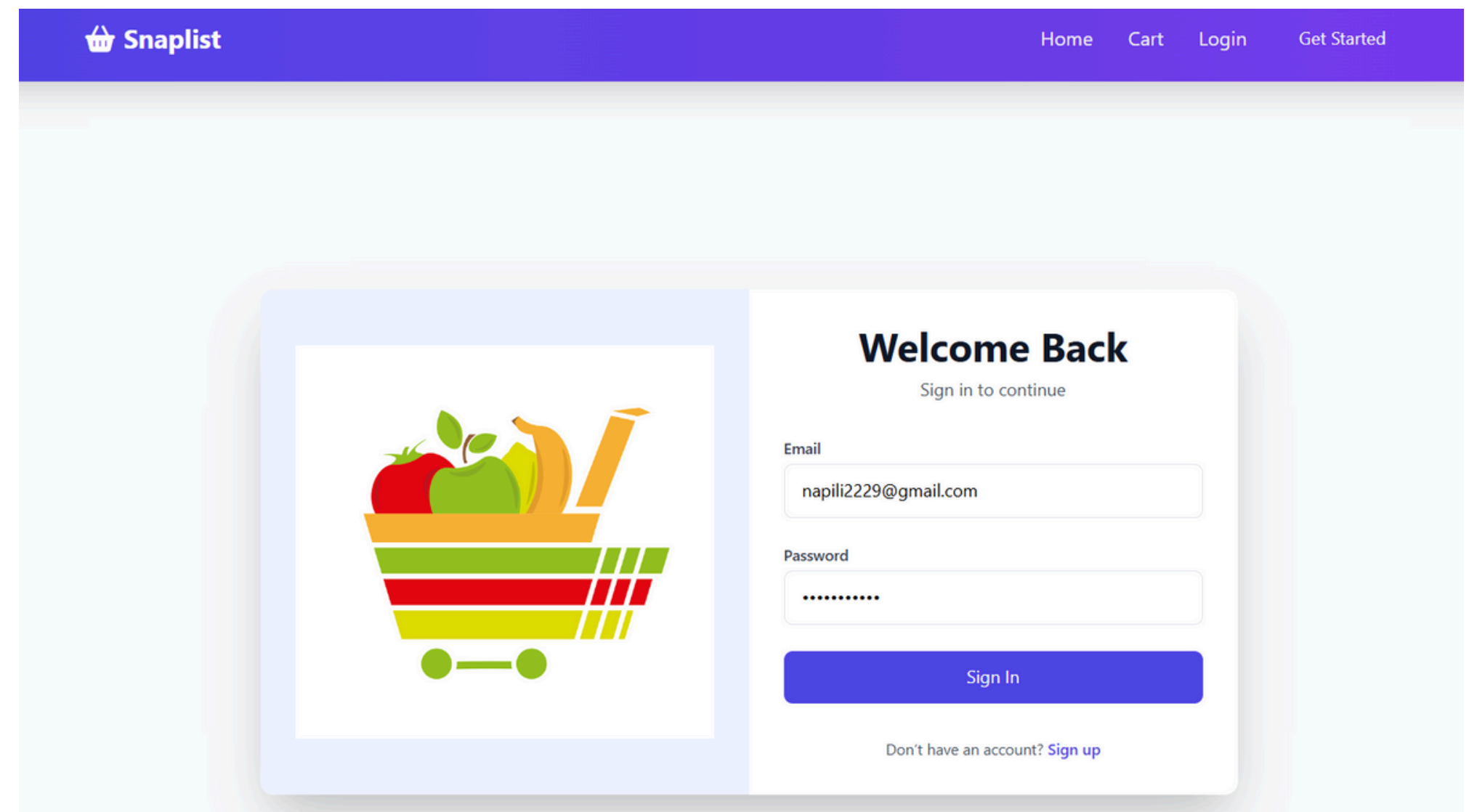


# Flask-Based Grocery Detection System

Go, change the world

- The Flask-based grocery detection system is an advanced web application designed to simplify the process of identifying and purchasing grocery items through cutting-edge image recognition technology.
- At its core, the system processes user-uploaded images to detect and classify grocery items using a MobileNet V3 trained model, a highly efficient deep learning architecture optimized for accurate and real-time object detection.
- The system seamlessly integrates image processing, real-time inference, and a database-driven shopping experience, ensuring smooth product recognition and cart management.

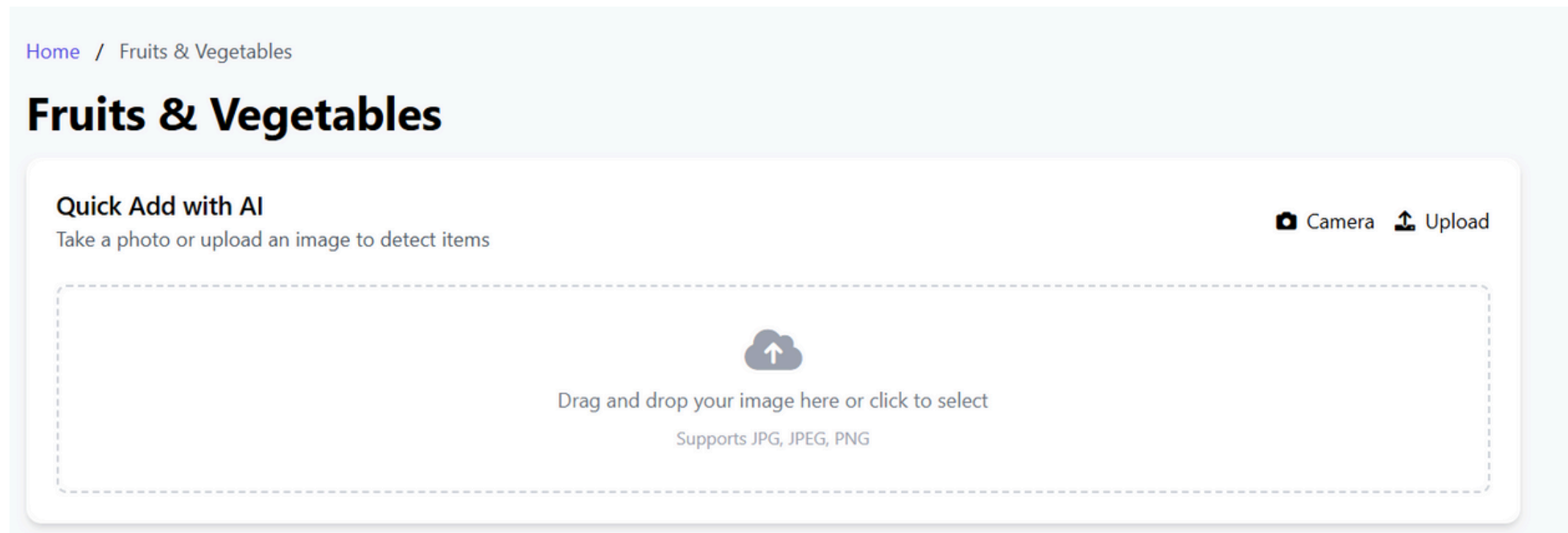
**User Registration and Login:** The registration and login processes were seamless, with users able to create accounts and log in without issues. The use of Flask-PyMongo for database management ensured that user data was securely stored and retrieved.





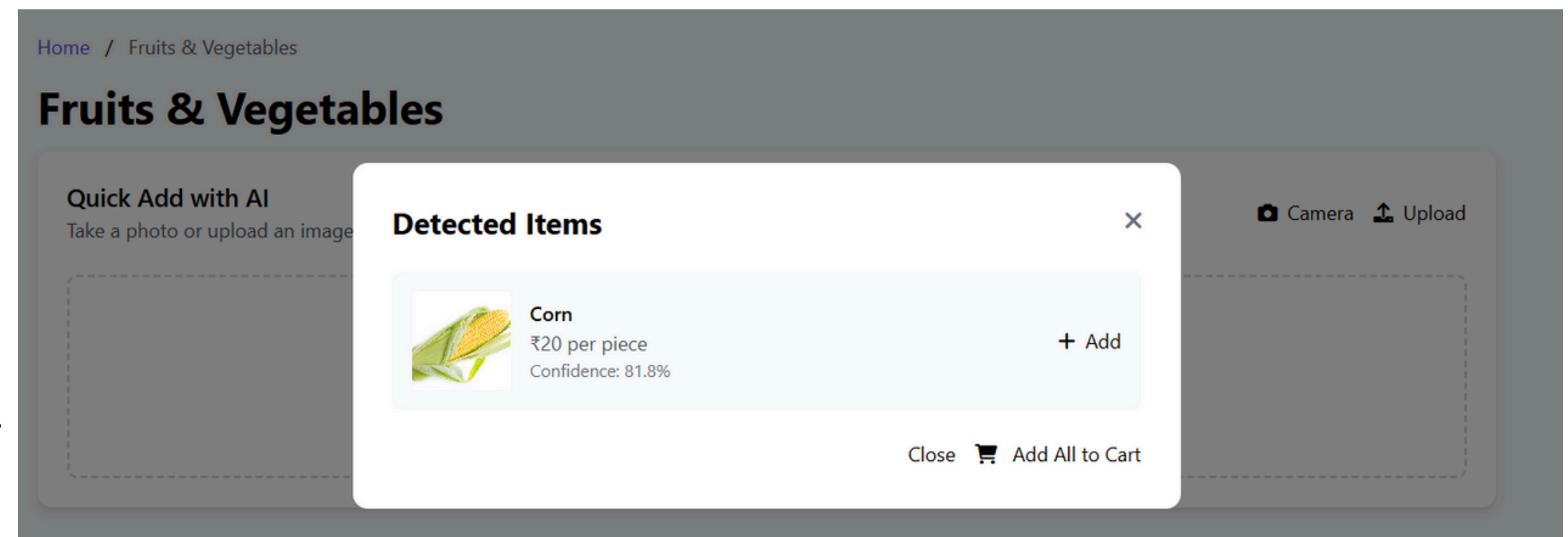
# Flask-Based Grocery Detection System

Go, change the world



- First, the image is received through the Flask web application and preprocessed using OpenCV.
- This includes resizing, noise reduction, and color normalization. Next, MobileNet V3 runs inference on the processed image to identify grocery items.

- The detected items are then cross-referenced with the MongoDB database to retrieve details such as name, price, and category.
- Finally, the system logs the processing time and detected items, ensuring efficient tracking and debugging of detections.




# Flask-Based Grocery Detection System

Go, change the world

- **Cart Management:** The cart functionality worked as expected, allowing users to add, update, and remove items. The cart total was dynamically updated, providing users with real-time feedback on their purchases.


## Shopping Cart

3 items




**Corn**  
20 INR per piece  
Category: Vegetables

– 1 +



**Banana**  
10 INR per piece  
Category: Fruits

– 1 +



**Chilli Pepper**  
80 INR per kg  
Category: Vegetables

– 1 +

Subtotal

110 INR

Delivery Fee

40 INR

Tax (5%)

6 INR

**Total**

**156 INR**

Proceed to Checkout

Continue Shopping



# Flask-Based Grocery Detection System

Go, change the world

- **Checkout Process:** The checkout process was straightforward, with users required to enter shipping and payment details. The application calculated the total cost, including taxes and delivery fees, and successfully processed order.

### Order Summary

Corn Qty: 1	20 INR
Cucumber Qty: 1	40 INR
Kiwi Qty: 1	50 INR
<hr/>	
Subtotal	110 INR
Delivery Fee	40 INR
Tax (5%)	6 INR
<hr/>	
<b>Total</b>	<b>156 INR</b>

### Shipping Information

Full Name

Phone Number

Address Line 1

Address Line 2 (Optional)

City  State

PIN Code

**Payment Method**

☐ Cash on Delivery

☐ UPI Payment



RV College of  
Engineering®

*Go, change the world*

**THANK YOU**