SMART VISION TECHNOLOGY QUALITY CONTROL

Flipkart GRiD 6.0 - Robotics Challenge

Flipkart

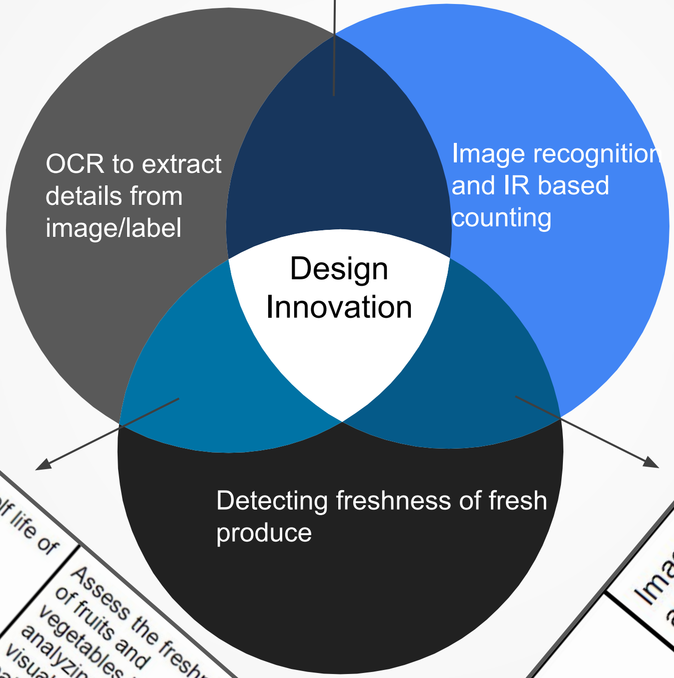
1.About us

We're **HAITech**, a group of three friends—**H**emashree, **A**swin, and **I**psita. We're currently in our 3rd year of B.Tech in Artificial Intelligence & Data Science From Shiv Nadar University Chennai .

kipsita205@gmail.com
https://www.linkedin.com/in/ipsita-kar-/

Use Cases Accomplished

Sl no	Use case	Weightage	Use case	Example
1	OCR to extract details from image/label	20%	Extracting details available on packaging material	Use of OCR to read brand details, pack size, brand name etc



4	Detecting freshness of fresh produce	40%	Predicting shelf life of fresh fruits and vegetables	Assess the freshness of fruits and vegetables by analyzing various visual cues and patterns

3	Image recognition and IR based counting	30%	Brand Recognition and count confirmation using IR

Index

OUR SOLUTION

- YOLO Object detection (Git Bash – YOLOV5) -15%
- Image Count detection – 15%
- OCR Text Extraction (StreamLit UI/UX + Kaggle NB + Dataset) – 20%
- Freshness Index ->(Fruits/Vegetables Classifier + Fresh vs Rotten + Index) – 40%
- TOTAL – 90% Use cases covered

Freshness of Produce:Proposal Overview

Objective: Develop a model to classify fresh and rotten fruits using deep learning.

Dataset: "Fruits Fresh and Rotten" dataset with images of various fruits (apples, oranges, bananas) in both fresh and rotten conditions.

Approach:

- Leverage **MobileNetV2** pre-trained model for image classification.
- Fine-tune the model for freshness detection and classification.

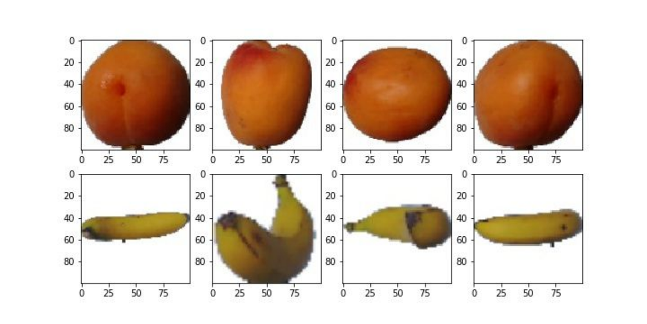
Data Pipeline & Preprocessing

Data Augmentation:

- Applied augmentation techniques like rotation, zoom, shift, and flip.
- Created **training** and **validation** splits.

Image Size: 150x150 pixels.

Batch Size: 64.



Preprocessing:

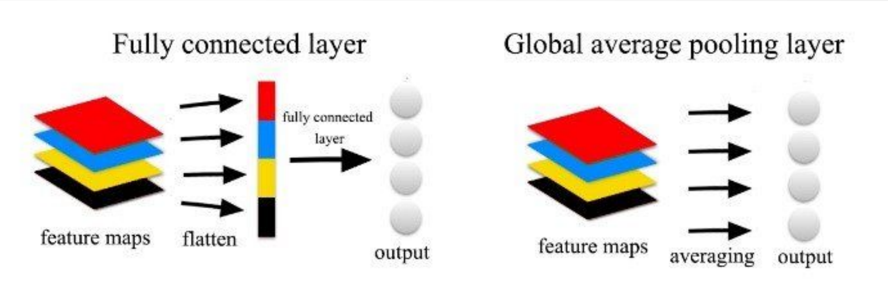
- Used `ImageDataGenerator` for augmentation and data loading.

Model Architecture & Training

Model: MobileNetV2 with fine-tuning on the last 20 layers.

Additional Layers:

- GlobalAveragePooling
- Dense layers (128 neurons)
- Dropout (0.5) to prevent overfitting.



Training:

- Optimizer: Adam with a learning rate of 1e-5.
- Early Stopping to avoid overfitting.
- Number of classes: Fresh and Rotten categories across fruits.

Results & Predictions

Model Performance:

- Achieved good accuracy after training on the augmented dataset.

Prediction:

- Freshness Index: A simple lookup for classifying fresh or rotten.
- Example: Test image of a banana classified as "freshbanana" with a freshness index of 'A'.

Next Steps:

- Improve model generalization.
- Test on new, unseen images.

A

FRESH

B

EDIBLE

C

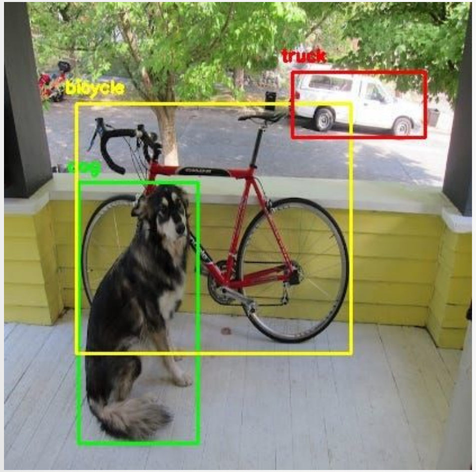
ROTTEN

Image Detection and Count:Proposal Overview

Objective: Detect and classify images using deep learning models.

Approach:

1. **Image Classification** using **MobileNetV2** for fruits and vegetables.
2. **Object Detection** using **YOLOv5** to count specified objects in images.



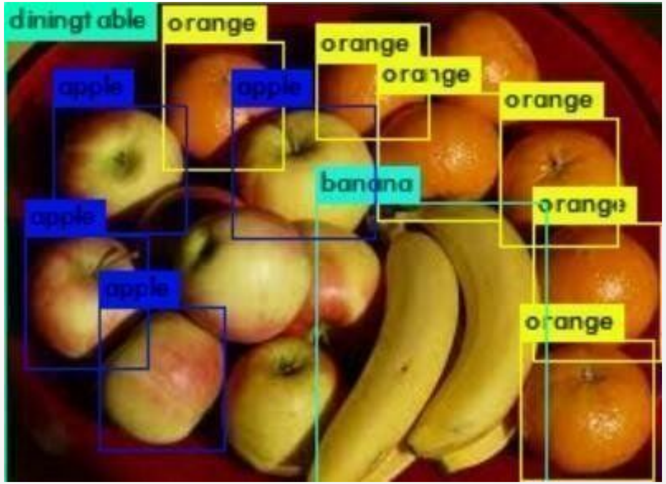
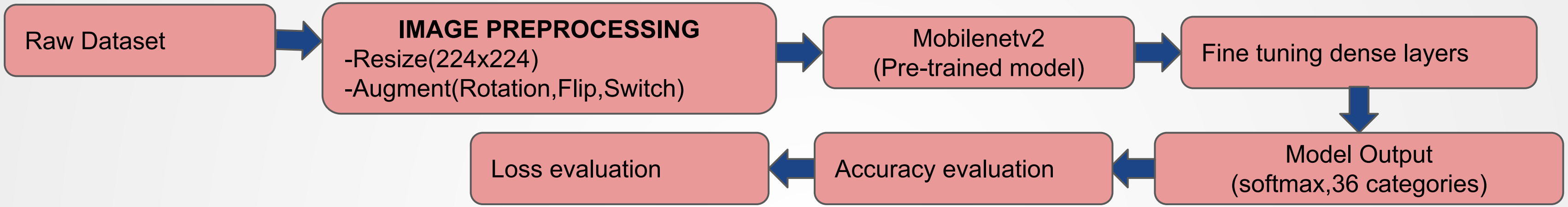
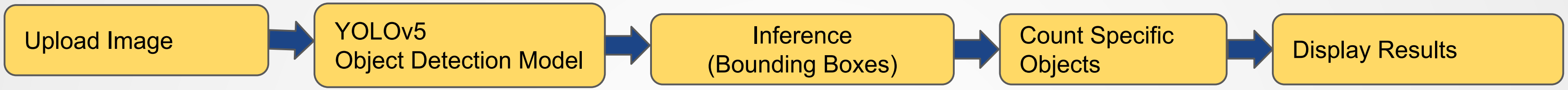


Image Classification Pipeline (Flowchart)



Step	Details
Input	Fruits & Veg Dataset (Train, Validation, Test)
Preprocessing	Resize to 224x224, Augmentation applied
Model	MobileNetV2 pre-trained
Output	36 categories (fruits/vegetables)

Object Detection Pipeline (Flowchart)



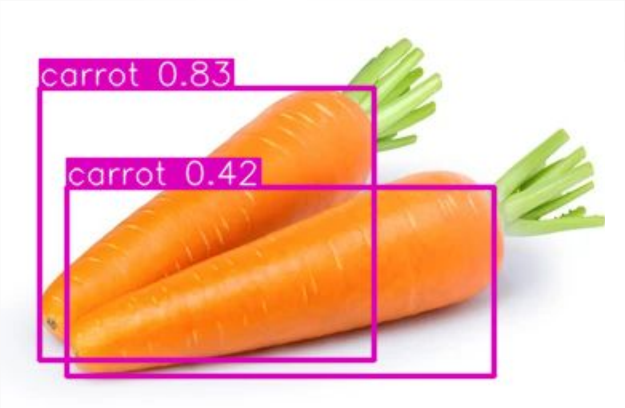
Step	Details
Input	Uploaded image (e.g., Fruits)
Model	YOLOv5 pre-trained model
Detection	Bounding boxes drawn around detected items
Output	Count of specified objects

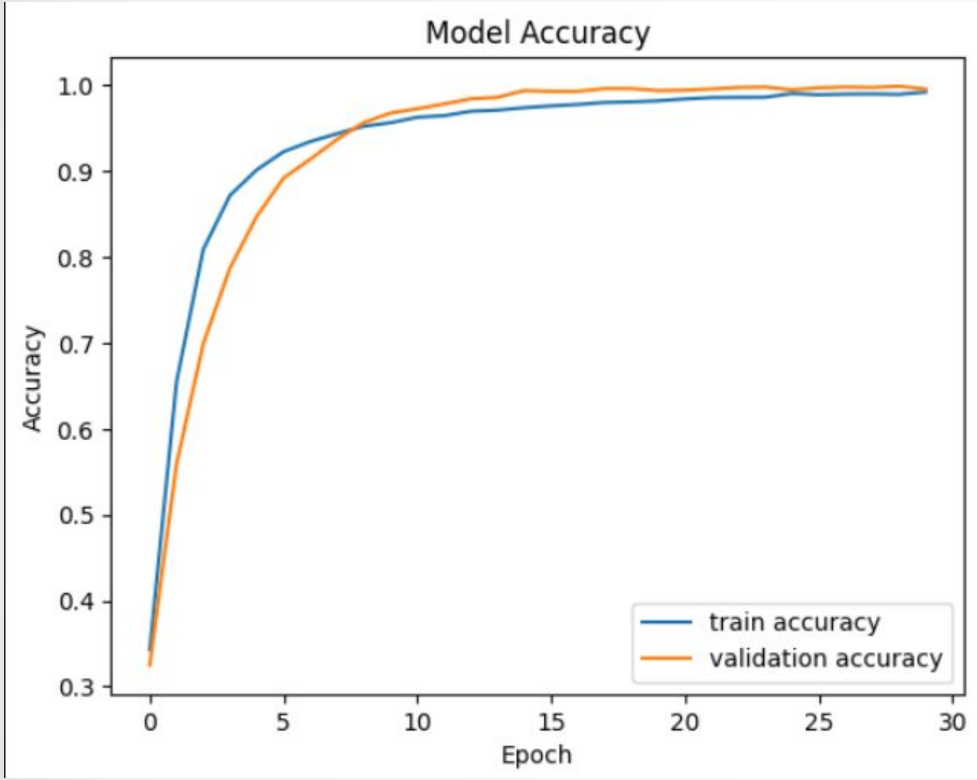
Results & Next Steps

Image Classification: Achieved reasonable accuracy in 5 epochs.

Object Detection: Successfully detected and counted objects using YOLOv5.

Output from our own Model.





It took extensive time period of 5 hours to train for 30 epochs.

Test Accuracy

We have trained different models with **5,15 and 30** epochs respectively:

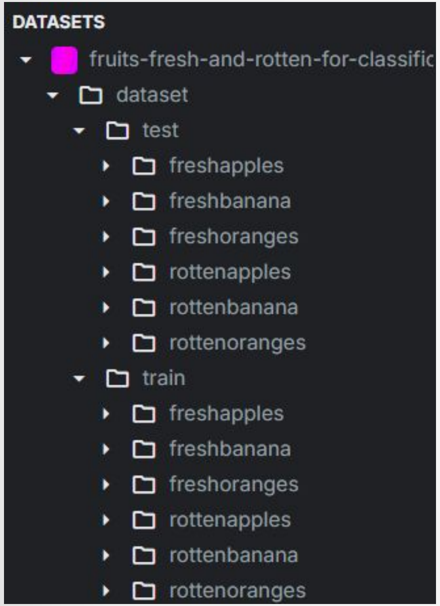
- Accuracy in 5 epochs-86.77%
- Accuracy in 15 epochs-82.36%
- Accuracy in 30 epochs-99.93%

The model demonstrates significant improvement in accuracy with increased training epochs, achieving near-perfect performance **(99.93%) at 30 epochs**, indicating that extended training allows the model to learn and generalize better, **although early overfitting is observed at 15 epochs** with a slight drop in accuracy.

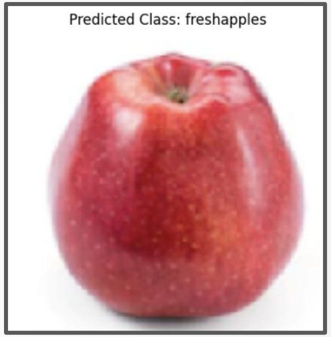
MODEL **REVIEW**

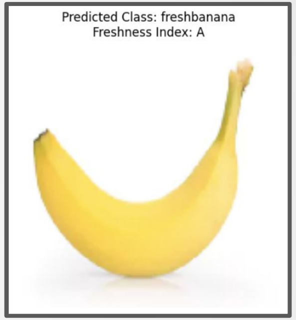
Grocery Freshness Index

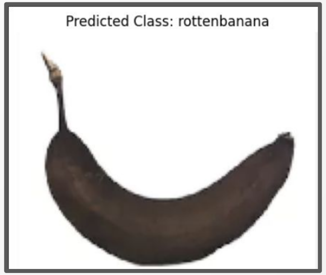
Dataset

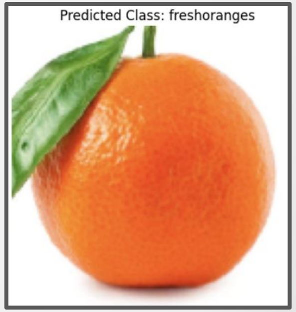


Outputs from our own model









OCR to extract details from image/label: Model Overview

Cost Efficiency:

- **Training Cost**: Minimal, as transfer learning was applied with pre-trained models, reducing GPU usage and training time.
- **Inference Cost**: Lightweight, with inference taking **<1 second** per image on GPU, ideal for real-time applications.

GPU Utilization: The model leverages a **NVIDIA Tesla P100** on Kaggle for faster OCR processing and classification tasks, significantly reducing inference time compared to CPU-based operations.

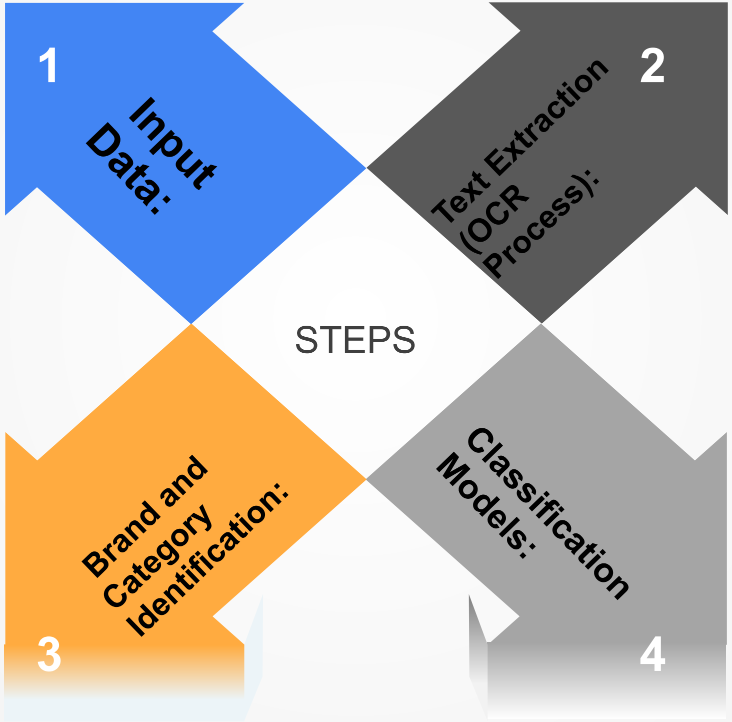
Accuracy: **90%** of products were correctly classified based on extracted text.
F1-Score: **0.87**, balancing precision and recall across categories.
Confusion Matrix:

- **True Positives**: Household (92%), Personal Care (88%)
- **False Negatives**: 7%
- **False Positives**: 5%

Data Pipeline Processing

- **Personal Care Products**: e.g., shampoos, creams, lotions, toothpaste.
- **Household Items**: e.g., detergents, disinfectants, food products.

- **Tool Used**: EasyOCR, PaddleOCR
- **Process**:
  1. Images are uploaded via the user interface (Streamlit).
  2. The EasyOCR tool is used to read the text from the product packaging.
  3. The `easyocr.Reader` function extracts raw text from the image.
  4. The extracted text is then cleaned and lowercased for uniformity.

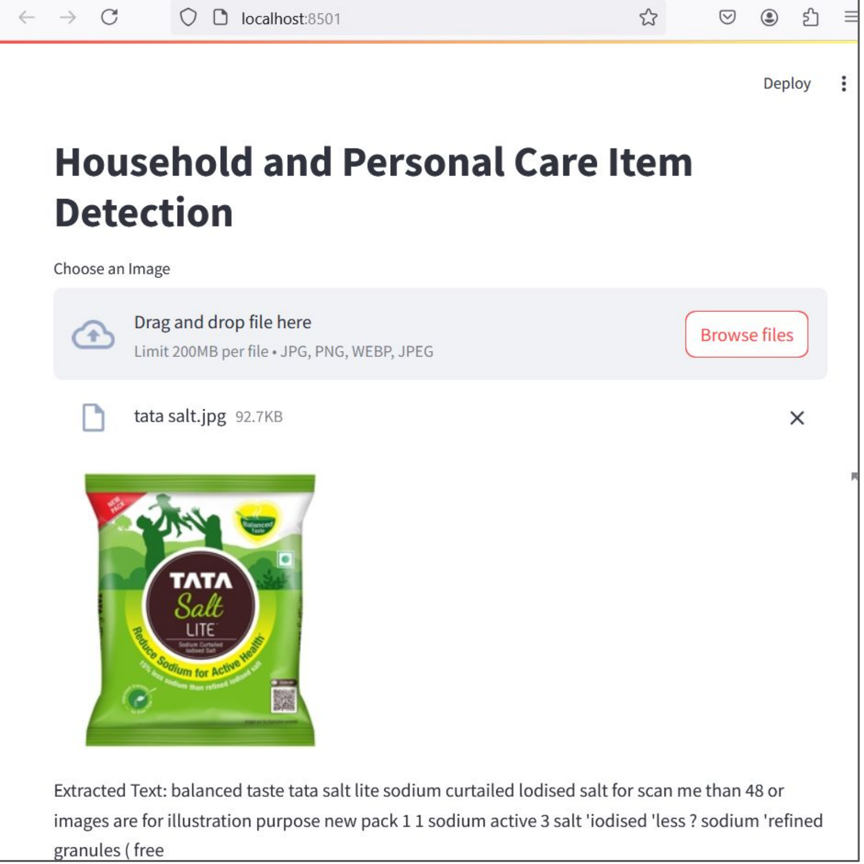


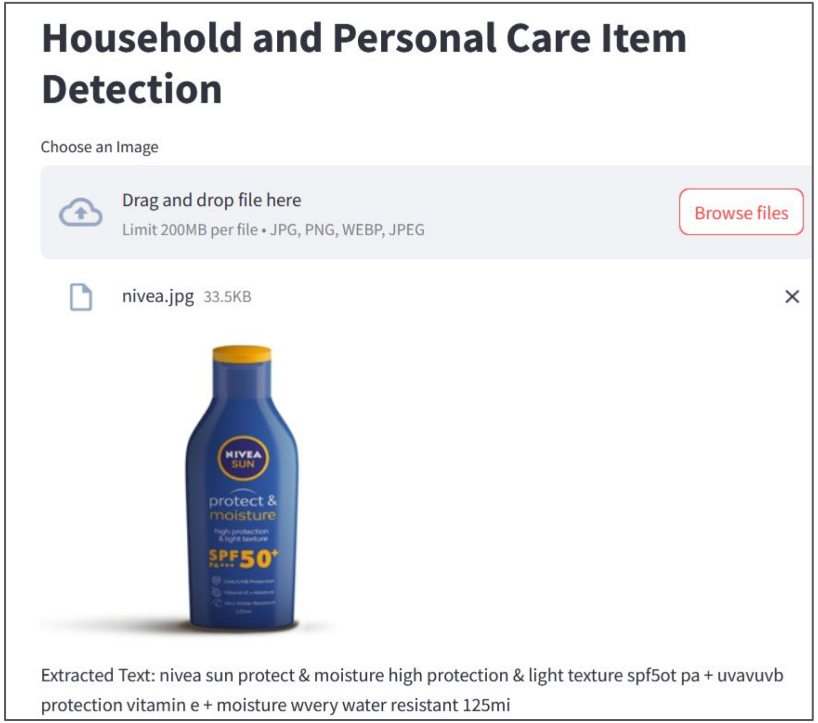
- The **extracted text** is matched against a dictionary (`brand_dict`) that maps known brand names (**e.g., Ariel, Dove, Colgate)** to their corresponding categories (e.g., **Household: Laundry Detergent, Personal Care: Oral Care).**
- This is done using Python's `difflib.get_close_matches` function to ensure a robust text match even when there are slight OCR misreads.

- **Models**:
  1. **Personal Care Classification Model**: `personal_care_model.pkl`
  2. **Household Classification Model**: `household_model.pkl`

These models take the extracted text as input and classify the product into specific subcategories or predict additional product attributes.

Outputs from our own model






```
Brand: TATA
Category: food
Other Details: Balanced , Taste, , , Lite, Sodium Curtailed, lodised , for, SCAN ME, than,
W, PACK, 1, 1, Sodium, Active, 3, , 'iodised, 'less ?, sodium, 'refined, Granules, (, Free
```

```
--- Extracted Information ---
Brand: Nivea
Category: skincare
Other Details: NIVEA, SPF5ot, 125mi
```

Model Architecture and Training

05 **Output Display**: The brand, category, prediction, and additional details are displayed to the user through the Streamlit interface.

04 **Model Prediction**: The pre-trained classification model predicts specific product details or subcategories (e.g., body wash, shampoo, detergent).

03 **Text Classification**: The relevant model (`personal_care_model` or `household_model`) is selected based on the identified category.

02 **Brand Identification**: The system compares extracted text with the brand dictionary to identify the product's brand and category.

01 **Image Upload**: User uploads an image of a product., **OCR Processing**: EasyOCR extracts text from the image.

4. Results and Predictions

Model Performance

- **Accuracy**: Achieved an accuracy of **90%**, indicating that 9 out of 10 products were correctly classified based on OCR extraction.
- **F1-Score**: An overall **F1-score of 0.87**, balancing **precision** and **recall**, particularly useful due to the imbalance between categories (more household than personal care items).

Predicted Output

- The model accurately predicted categories for unseen products:
  - **Personal Care**: e.g., **Dove (Body Wash)**, **Nivea (Skin Care)**.
  - **Household**: e.g., **Ariel (Laundry Detergent)**, **Amul (Dairy Products)**.
- Differentiated products like **Amul** (Household Food) and **Dove** (Personal Care Body Wash).

Streamlit-Based User Interface

- **Image Upload**: Users upload product images (supported formats: **jpg, png, webp**).
- **Text Extraction**: The app extracts and displays text within **2-3 seconds**.
- **Brand & Category Detection**: Displays brand, category, and additional product details.
- **Product Information**: For food products, details like **nutritional information** are fetched from Google in **<5 seconds**.

THANK YOU!!
DONE BY:
IPSITA KAR
HEMASHREE RAMESH BABU
ASWIN CHANDRASEKAR