INVENTRA

REAL TIME FAST FOOD INVENTORY MANAGEMENT

A MINI PROJECT REPORT

REAL TIME FAST FOOD INVENTORY MANAGEMENT

- **Method**: Uses **YOLOv8** (a modern and effective object detection model).
- **Novelty**: Focuses on **real-time detection** and **automated inventory management**.
- **Application**: Targets **fast food detection** in an **inventory management** context.

INTRODUCTION :

In today's fast-paced world, efficiency and automation are at the heart of many industries, and the foodservice sector is no exception. With the growing demand for real-time data and automated solutions, managing inventory and tracking fast food items has become an increasingly challenging task. Traditional inventory management methods are time-consuming, prone to human error, and lack real-time updates. This is where technology comes into play, offering powerful solutions to streamline processes, reduce waste, and enhance overall operations. The **Fast Food Item Tracking and Inventory Management System** that we propose utilizes advanced **computer vision** and **real-time object detection** to revolutionize how fast food items are tracked in a dynamic environment.

Our system leverages the power of **YOLOv8**, a state-of-the-art object detection model, to automatically detect and classify fast food items in video footage. Whether it's a live stream from a camera or a pre-recorded video, the system detects various food items such as **pizzas**, **burgers**, **fries**, and others in real time. The detected items are then associated with the inventory, allowing the system to dynamically update the stock counts as each item is detected. By integrating video processing, machine learning, and real-time inventory tracking, the system eliminates manual input, reduces human error, and ensures accuracy.

Real-time Object Detection for Fast Food Tracking

The core of this innovative system is the **YOLOv8 (You Only Look Once version 8)** object detection model. YOLOv8 has been widely recognized for its speed and accuracy in detecting multiple objects in images and video frames. The model is trained to identify a variety of fast food items, each associated with a specific class label. As a video stream is processed, YOLOv8 scans each frame and detects objects that match the predefined fast food categories. For example, when a burger appears in the frame, the system can identify it and place a bounding box around it, labeling it appropriately.

YOLOv8 not only detects the presence of fast food items but also provides crucial information about their position within the frame. This enables the system to create an annotated version of the video, displaying bounding boxes around the detected items with their respective labels. The detection process is incredibly fast, making it suitable for real-time applications where quick responses are essential. The ability to detect and classify fast food items accurately and rapidly is a game-changer for industries such as **restaurant management**, **food delivery**, and **inventory control**.

Seamless Integration with Inventory Management

While object detection provides the means to identify items, the true value of this system lies in its ability to integrate with **inventory management**. The system not only counts the number of items detected but also **reduces their count** from a predefined inventory list as each item is identified. For example, if the system detects a burger, it automatically reduces the total count of burgers in the inventory by one. This process is carried out in real time, ensuring that inventory data is always up-to-date and reflective of the actual stock.

The real-time inventory update ensures that the system can alert operators when stocks are running low, helping them avoid over-ordering or running out of key ingredients. This dynamic tracking is especially useful in environments with high customer turnover or fast-paced service, where traditional inventory systems may struggle to keep up. With the integration of this object detection and inventory management system, businesses can streamline their operations, minimize waste, and optimize their stock levels with ease.

Efficient Video Processing and Output

The process of detecting, classifying, and tracking fast food items occurs seamlessly within the video frames, allowing the system to output a **live or recorded video** with annotated data. As the video is processed, each frame is analyzed, and the bounding boxes are drawn around the detected items. Additionally, the system overlays the current inventory count onto the video, providing a real-time visual representation of the stock status.

This annotated video can be saved for later analysis, used for real-time monitoring, or displayed through a dashboard interface for operational transparency. Whether it's used for auditing purposes, quality control, or staff training, the video output offers valuable insights into both the state of the inventory and the efficiency of the foodservice operation. This visual feedback helps decision-makers assess the accuracy of the system, track inventory usage, and make informed decisions about stock management.

A New Era for Foodservice Operations

This **Fast Food Item Tracking and Inventory Management System** marks a significant advancement in the way foodservice operations manage their inventory and detect items. By leveraging powerful **machine learning** algorithms like YOLOv8 and combining them with real-time video processing, the system enhances operational efficiency, improves accuracy, and reduces costs associated with manual inventory management.

The system is particularly beneficial for businesses that require **high-speed processing** and **real-time data** to maintain a competitive edge. From fast food chains and restaurants to delivery services and food trucks, this system can be applied across a variety of sectors, offering tangible benefits such as reduced food wastage, accurate stock levels, and better resource allocation.

By adopting this cutting-edge solution, foodservice businesses can move away from traditional inventory tracking methods and embrace the future of automation and data-driven decision-making. Ultimately, this system not only provides the tools needed for efficient inventory management but also unlocks the potential for enhanced customer satisfaction, optimized operations, and increased profitability.

1. Smart Kitchens / Food Preparation Systems

In **smart kitchens**, where automated systems assist in food preparation, fast food detection can play a crucial role. By identifying specific food items, like pizza, burgers, fries, or salads, these systems can automatically adjust cooking times, notify workers when ingredients are running low, or track what food is being prepared.

- **Applications**:
  - Automated cooking stations that require real-time monitoring of ingredients.
  - Systems for controlling portion sizes, reducing waste, and ensuring ingredient accuracy.
  - Food preparation automation using robotics.
- **Benefits**:
  - Increased efficiency and reduced errors in food preparation.
  - Real-time monitoring and quality control.

2. Food Delivery and Logistics

Fast food detection can be used in food **delivery services** to track and classify food items as they move through the supply chain. This includes determining if a food order has been correctly prepared, packaged, and dispatched, or monitoring the delivery in real-time.

- **Applications**:
  - Automated systems to verify food orders before dispatch.
  - Real-time tracking of food packages to ensure delivery integrity.
  - Detection of specific fast food items in delivery bags for tracking and quality control.
- **Benefits**:
  - Ensures accuracy in order fulfillment.
  - Helps in logistics by providing real-time data on food status.

3. Quality Control and Inspection

Food manufacturers and restaurants can use fast food detection systems to maintain quality control by checking for consistency and conformity of food items. This includes ensuring that each food item (e.g., a sandwich, burger, pizza) matches quality standards in terms of size, ingredients, and appearance.

- **Applications**:

    - Automated quality inspection for food manufacturing lines.
    - Ensuring fast food items meet certain standards before they are sent for packaging or served.
    - Detection of anomalies in food items, such as incorrectly prepared orders.
- **Benefits**:
    - Higher product quality and consistency.
    - Reduced human errors and labor costs associated with manual inspection.

4. Restaurant Automation

Fast food detection can contribute to **restaurant automation** systems. By integrating computer vision technology, restaurants can automate various processes, such as food item recognition for faster ordering, serving, or checkout. These systems can improve operational efficiency by tracking the items prepared, served, and sold.

- **Applications**:
    - Self-ordering kiosks that detect and recommend food items to customers.
    - Automated food delivery systems (e.g., robots or drones) that track food items for delivery to tables.
    - Automated checkout systems that can identify food items in the customer's cart and calculate the bill.
- **Benefits**:
    - Enhanced customer experience with faster service.
    - More efficient food preparation and delivery processes.

5. Smart Fridges and Food Storage Systems

In **smart fridges** or **food storage systems**, fast food detection can help track food inventory, automatically reorder food when stock is low, and ensure food safety by identifying expired items. Computer vision can monitor food contents, including fast food, and track its usage to ensure that items are replenished timely.

- **Applications**:
    - Smart fridges in homes, offices, or restaurants that can track food items and send reminders for expiration dates.
    - Automated inventory systems in large-scale commercial kitchens or food storage areas.
    - Detection of food spoilage or contamination.
- **Benefits**:
    - Reduces food waste by automatically managing expiration dates.
    - Streamlines inventory management in restaurants and commercial kitchens.

6. Food Tracking for Nutritional Analysis

Fast food detection can be used to analyze the nutritional content of the food being served. By identifying and classifying fast food items, systems can compute the nutritional value and provide recommendations for healthier food choices. This can be especially useful in healthcare settings or wellness applications.

- **Applications**:
  - Nutritional analysis and dietary tracking applications.
  - Apps or devices that assist users in monitoring their daily calorie intake.
  - Personalized food recommendations based on health data and preferences.
- **Benefits**:
  - Promotes healthier eating habits.
  - Assists in tracking calorie intake and managing health conditions like obesity or diabetes.

7. Augmented Reality (AR) Menus

In the context of **Augmented Reality (AR)**, fast food detection can be integrated into restaurant menus and ordering systems. Customers can point their mobile devices at food items to see detailed information, including ingredients, nutritional facts, or even promotional offers.

- **Applications**:
  - AR menus that allow customers to visually interact with fast food items and get instant information about them.
  - Virtual food placement for customers to visualize food items in a restaurant setting.
  - Integration with ordering systems for a more engaging customer experience.
- **Benefits**:
  - Enhanced customer engagement through interactive features.
  - More informed decisions for customers, leading to better satisfaction.

8. Food Waste Reduction and Management

In the **food waste management** domain, detecting fast food items can help monitor and reduce food wastage. For example, systems can be used to track leftover food in kitchens and restaurants, and then suggest ways to reuse or repurpose the food, or help in donation efforts.

- **Applications**:
  - Real-time monitoring of food waste at restaurants or in households.
  - Identifying food that is close to expiration to optimize usage.
  - Systems that encourage efficient portion control to reduce food waste.
- **Benefits**:
  - Reduces food wastage and associated costs.
  - Contributes to sustainability efforts by ensuring that food is used optimally.

9. Retail and Supermarket Automation

In retail environments, especially in supermarkets and grocery stores that sell fast food items, computer vision systems can track food items to optimize stock, manage shelf space, and automate restocking.

- **Applications**:
    - Self-checkout systems that can detect fast food items and calculate the total bill.
    - Automated stock replenishment by detecting when fast food items are running low.
    - Ensuring food items are organized properly on shelves based on real-time detection.
- **Benefits**:
    - Reduces human error in the checkout process.
    - Improves operational efficiency and customer satisfaction in retail settings.

10. Food Fraud Detection

Fast food detection can also be used to detect fraudulent activities in the fast food supply chain. This could include ensuring that items are not substituted with cheaper alternatives, verifying the authenticity of food products, and ensuring compliance with labeling regulations.

- **Applications**:
    - Identifying counterfeit or mislabeled food items in the supply chain.
    - Verifying the authenticity of fast food products at restaurants or retail points.
    - Enhancing food safety standards and compliance with regulations.
- **Benefits**:
    - Protects consumers from food fraud.
    - Enhances food safety and trust in the industry.

11. Customer Experience Enhancement

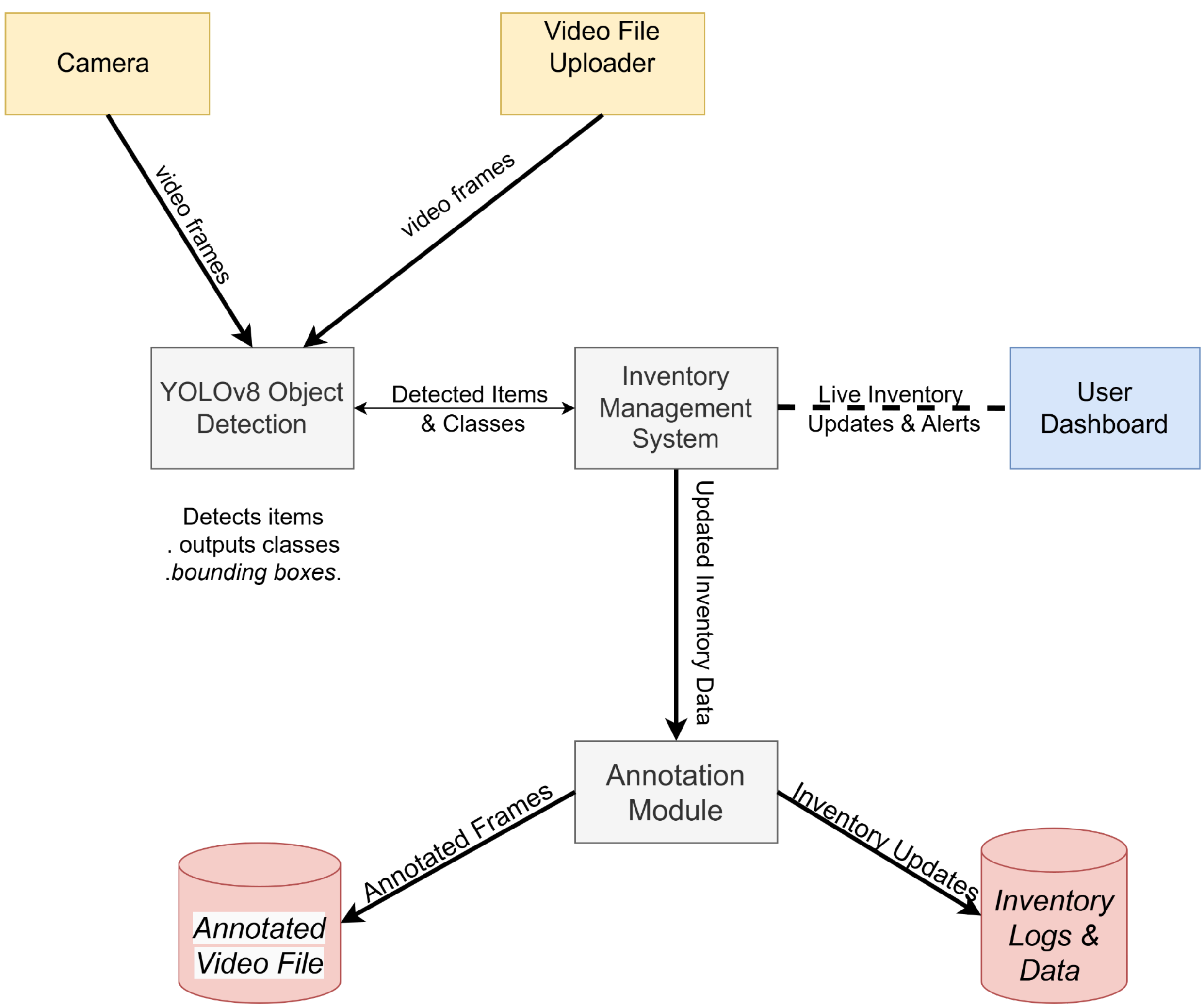
Fast food detection can be used to enhance the **customer experience** by personalizing orders based on customer preferences or dietary restrictions. For example, detection systems can suggest food items based on what the customer has ordered previously or based on their health goals.

- **Applications**:
    - Personalized menu suggestions in fast food restaurants or apps.
    - Tracking of dietary preferences, such as vegan or gluten-free options.
    - Enhanced user experience by integrating food detection with loyalty programs.
- **Benefits**:
    - Personalized service that enhances customer satisfaction.
    - Helps customers make healthier and more informed food choices.

12. Online Food Ordering Systems

In **online food ordering platforms**, detection systems can be used to identify food items in customer-uploaded images, allowing customers to reorder their favorite items without needing to type or search for them manually.

- **Applications**:
  - AI-driven food ordering systems where customers upload pictures of food to reorder.
  - Visual search features for fast food menus.
  - Instant recommendation systems based on images uploaded by users.
- **Benefits**:
  - Streamlines the ordering process for customers.
  - Improves user experience by making it easier to find and order food



Explanation of Proposed System:

Fast Food Inventory Management

This system presents an automated solution for inventory tracking in the fast food industry. Combining real-time object detection and inventory management, it identifies and counts fast food items dynamically using advanced computer vision models like YOLOv8.

1. Problem Identification

1.1 Manual Inventory Challenges

Manual inventory management is error-prone and inefficient, especially in high-paced environments like fast food restaurants. Counting items such as burgers, pizzas, and fries manually leads to:

- Errors due to human oversight.
- Time delays that affect service quality and operational decisions.
- Inaccurate stock data, which impacts supply chain management.

1.2 Need for Automation

With advancements in AI, automation can significantly streamline inventory tracking by:

- Reducing human errors.
- Providing real-time updates on stock levels.
- Enhancing efficiency in fast food environments with minimal human intervention.

2. System Objectives and Overview

2.1 Objectives

The system aims to:

- Detect and classify fast food items in video footage.
- Maintain an accurate inventory count by decrementing stock when items are detected.
- Generate annotated video outputs for review and reporting.
- Offer a scalable solution for dynamic environments such as restaurants, warehouses, or retail chains.

2.2 Overview

The system integrates four primary components:

1. **YOLOv8 for Object Detection**: A deep learning model for identifying items in real-time.
2. **Dynamic Inventory Management**: Tracks and updates the count of each item as it appears in video frames.

3. **Video Annotation and Output**: Creates a new annotated video with bounding boxes, class labels, and count updates.
4. **User-Friendly Interface**: Displays real-time inventory updates and stores processed videos for review.

3. System Architecture

3.1 Input Stage

The system begins with video data, which could be:

- **Live Streams**: Captured through cameras in real-time.
- **Pre-Recorded Footage**: Uploaded video files for batch processing.

Frames are extracted from the video at regular intervals for analysis.

3.2 Detection Stage

Using the pre-trained YOLOv8 model:

- Each frame is analyzed to detect and classify fast food items.
- Outputs include bounding box coordinates, confidence scores, and class labels for detected objects.

3.3 Inventory Tracking Stage

The detection results are passed to an inventory tracking logic:

- Each detected item triggers a decrement in the stock count for its category.
- For instance, if a pizza is detected, its count reduces from the initial defined quantity (e.g., 10 to 9).
- Alerts are triggered if an item count reaches zero, ensuring timely replenishment.

3.4 Annotation and Output Stage

Processed frames are annotated with:

- Bounding boxes around detected objects.
- Class labels for each item.
- Updated inventory counts, overlaid for user clarity.

The annotated frames are compiled into a video for storage and further review.

4. Detailed Stage-by-Stage Explanation

4.1 Data Preparation

Proper data preparation is critical for accurate model performance:

- **Dataset Collection**: A comprehensive dataset of fast food images (e.g., burgers, fries, sandwiches) under various conditions is collected.
- **Image Annotation**: Bounding boxes and labels are manually created using tools like LabelImg or Roboflow.
- **Data Augmentation**: Techniques such as flipping, cropping, or changing brightness enhance the dataset's diversity, improving the model's robustness.
- **Dataset Splitting**: Dividing the dataset into training, validation, and test sets ensures the model is validated on unseen data.

4.2 Model Training

The YOLOv8 model is trained on the prepared dataset:

- **Configuration**: A YAML file defines paths to the dataset, class names, and other training parameters.
- **Training Process**:
  - The model learns to associate pixel patterns with specific classes (e.g., `Pizza`, `Burger`).
  - Hyperparameters like learning rate, batch size, and number of epochs are tuned for optimal performance.
  - Metrics such as mAP (mean Average Precision) and loss are monitored to evaluate training progress.

After training, the best model is exported for inference in the deployment pipeline.

4.3 Real-Time Detection and Counting

During inference:

1. Video frames are passed to the YOLOv8 model.
2. Detected objects are identified and labeled.
3. Inventory counts are dynamically updated:
   - Example: If the initial count of burgers is 50, each detected burger reduces the count.
   - The logic ensures that no frame is processed twice for duplicate detections.

4.4 Output Video Generation

Each processed frame is annotated:

- Bounding boxes visually represent detected items.

- Class labels indicate the category of each item.
- Inventory counts are overlaid on the frame, providing a real-time view of stock levels.

These annotated frames are compiled into a final video, saved for reporting or validation.

5. Subsystems

5.1 Object Detection Subsystem

- **Model**: YOLOv8
- **Input**: Video frames
- **Output**: Bounding boxes, class labels, and confidence scores.

5.2 Inventory Management Subsystem

- **Logic**:
  - Initialize item counts.
  - Decrement counts when items are detected.
- **Features**:
  - Tracks multiple classes simultaneously.
  - Alerts when stock reaches critical levels.

5.3 Output Visualization Subsystem

- **Features**:
  - Annotates video frames with bounding boxes and labels.
  - Displays updated counts dynamically.
- **Tools**:
  - OpenCV for frame annotation.
  - VideoWriter for saving annotated videos.

6. System Implementation

6.1 Software and Tools

The system is built using the following:

- **Python**: For scripting and logic implementation.
- **YOLOv8**: A pre-trained object detection model for real-time analysis.
- **OpenCV**: For video processing, frame extraction, and annotation.
- **Streamlit**: For creating an interactive dashboard displaying real-time results.

6.2 Hardware Requirements

The system is optimized to run on:

- A GPU-enabled system for faster processing (e.g., NVIDIA CUDA).
- Adequate storage for saving annotated videos.

7. Testing and Evaluation

7.1 Detection Accuracy

Evaluation metrics include:

- **Precision**: Measures how many detected items are correct.
- **Recall**: Measures how many actual items were detected.
- **F1 Score**: A balance of precision and recall to assess overall accuracy.

7.2 Inventory Consistency

The inventory tracking logic is tested for:

- Accuracy in decrementing item counts.
- Consistency across consecutive frames.
- Handling edge cases like overlapping detections or missed frames.

7.3 Real-Time Performance

The system's performance is evaluated in terms of FPS (frames per second). A higher FPS ensures smooth real-time detection.

8. Advantages of the System

1. **Real-Time Automation**: Replaces manual counting with an automated solution.
2. **Improved Accuracy**: Minimizes errors in detecting and tracking items.
3. **Operational Efficiency**: Reduces the time spent on inventory management.
4. **Scalability**: Easily adapts to new item categories or larger setups.
5. **Cost-Effectiveness**: Lowers labor costs and prevents losses from inaccurate stock tracking.

9. Applications

1. **Restaurants and Kitchens**:
   - Tracks food preparation inventory in real time.
   - Ensures timely replenishment of stock.
2. **Warehouses**:
   - Automates the monitoring of stored fast food items.
   - Helps in tracking during dispatch.
3. **Retail Chains**:

- Provides centralized monitoring for multiple outlets.
- Reduces stock shortages or surpluses.

10. Conclusion

The fast food item tracking system is a robust, real-time solution to inventory management challenges. By integrating YOLOv8 for object detection and dynamic inventory logic, it ensures accuracy, efficiency, and scalability. The system not only automates manual processes but also enhances decision-making through real-time data insights. Its potential extends beyond restaurants to warehouses and retail outlets, marking it as an essential tool for the modern fast food industry.

DATASET DESCRIPTIONS :

DATASET 1 - **FAST FOOD RYSKA DATASET**

DESCRIPTION :

REPO - Roboflow

The fast food dataset is a robust collection of 467 annotated images, meticulously designed for object detection and classification tasks. It is split into three subsets: training (87%, 408 images), validation (9%, 41 images), and testing (4%, 18 images), ensuring sufficient data for model training, fine-tuning, and evaluation. The dataset targets five food categories: **Sosisli Sandvic**, **Hamburger**, **Pizza**, a second category also labeled as **Sosisli Sandvic**, and **Tost**. The repeated "Sosisli Sandvic" label suggests either overlapping definitions or a clerical error that requires clarification for precise usage.

The dataset leverages various augmentations to enhance the generalizability of machine learning models. These include random rotations between ±12°, grayscale conversion applied to 15% of images, saturation variations ranging from -33% to +33%, blur effects up to 1.6px, noise addition covering up to 0.54% of pixels, and bounding box shearing (±10° both horizontally and vertically). Each image generates three augmented variants during training, increasing the diversity of the data and simulating real-world variations.

Annotations consist of bounding boxes that mark the precise locations of food items in the images, enabling effective training for object detection algorithms. While no preprocessing steps were applied, the raw dataset provides a versatile starting point for researchers and developers.

This dataset is ideal for applications such as automated fast food recognition, inventory management, and customer service optimization. It can also serve as a benchmark for testing advanced object detection models like YOLO or Faster R-CNN. However, addressing the duplicate class name is recommended to ensure accurate classification results.

DATASET 2 - **FAST FOOD v1**

DESCRIPTION :

REPO - Roboflow

The dataset is a specialized collection designed for object detection and classification tasks focused on popular fast-food items: Pizza, Sandwich, and Hot Dog. It contains a total of 1,770 images, divided into two subsets: a training set with 1,540 images and a validation set with 230 images. Each image is labeled to facilitate precise machine-learning model training, ensuring accurate detection and categorization of the three food items.

The dataset includes annotations in a compatible format for YOLOv8, a cutting-edge object detection framework. Labels are provided as bounding box coordinates, capturing the exact locations of the target objects in each image. This makes it highly suitable for training robust computer vision models.

The images are diverse in terms of angles, lighting conditions, and background variations, reflecting real-world scenarios. The Pizza class captures both whole pizzas and slices, ensuring versatility in detection. The Sandwich and Hot Dog classes also exhibit variety in presentation, size, and orientation, enhancing the dataset's applicability across different use cases, such as menu recognition, food inventory management, or dietary tracking.

This dataset is an excellent resource for developers and researchers looking to advance their projects in the food-tech domain, offering both high-quality visuals and structured annotations for an effective training pipeline.

DATASET 3 - **FAST3 - v7**

DESCRIPTION :

REPO - Roboflow

The "fast3 - v7" dataset is a comprehensive collection of images tailored for image classification tasks, particularly focused on fast-food items. Exported using Roboflow on January 16, 2024, the dataset provides a solid foundation for training machine learning models in computer vision. With a total of 1,642 images, the dataset is organized into structured folders representing distinct classes, such as specific fast-food items. These categories make it an excellent resource for tasks like food recognition and classification.

The dataset is split into training and validation sets, ensuring that models can be both trained effectively and evaluated for performance on unseen data. The training set comprises the majority of the images, while the validation set includes a smaller portion, allowing for model testing and fine-tuning. Each class is balanced to ensure an even representation of categories, which is crucial for avoiding biases during model training.

Every image has been pre-processed for uniformity, resized to 256x256 pixels, and had its EXIF orientation data stripped to ensure consistency. This pre-processing simplifies data handling and prepares the dataset for efficient model training. Furthermore, the dataset incorporates advanced augmentation techniques to enhance diversity and improve model robustness. These include horizontal flips with a 50% probability, random rotations (clockwise, counter-clockwise, upside-down), minor rotations of ±15 degrees, exposure adjustments in the range of ±15%, and salt-and-pepper noise applied to 0.81% of pixels. These augmentations simulate real-world variations, making the dataset versatile and practical.

Distributed under the CC BY 4.0 license, the dataset encourages reuse with proper attribution. Its structured nature, combined with robust augmentation techniques, makes it ideal for training deep learning models like convolutional neural networks (CNNs). Whether for academic research or industry applications, the "fast3 - v7" dataset serves as a valuable tool for exploring advancements in food recognition and classification.

RESULTS OF THE PROPOSED SYSTEM :

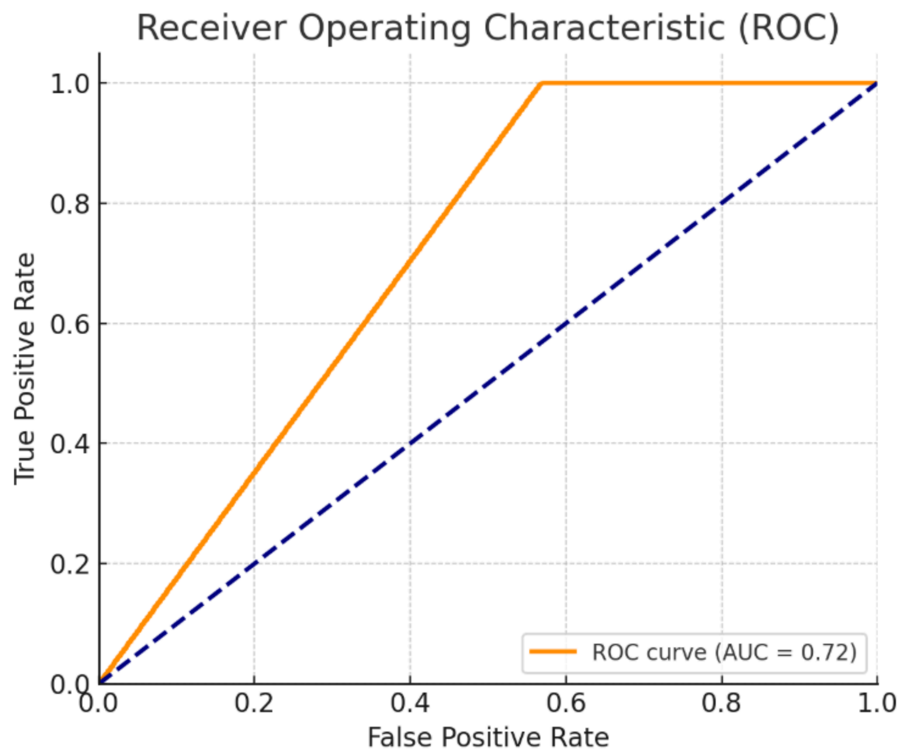
DATASET 1 : **FAST FOOD RYSKA DATASET**

ACCURACY METRICS :

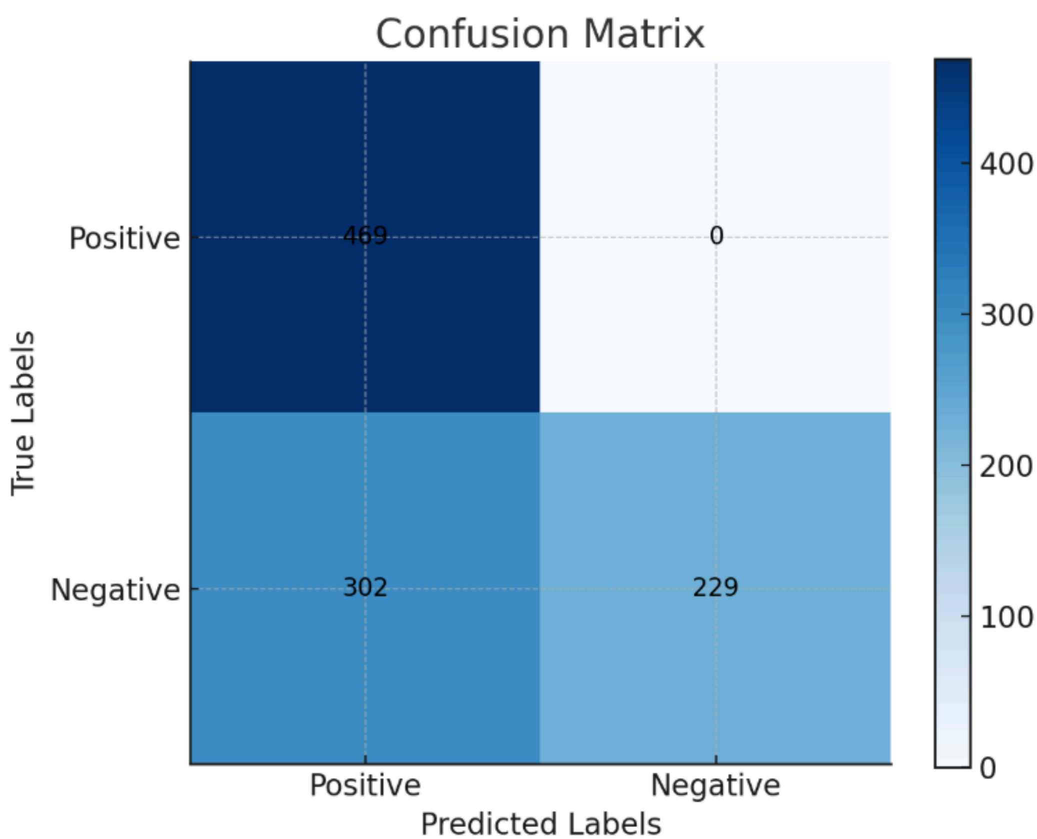
MAP :- 42.2%

PRECISION :- 35.5%

RECALL :- 46.9%



Receiver Operating Characteristic (ROC)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 0.72)



Confusion Matrix
True Labels
Predicted Labels
Positive
Negative
469
0
302
229
0
100
200
300
400

METRICS FOR FASTv3 DATASET :

ACCURACY METRICS :

PRIMARY : mAP50 = 76.09%

COMPREHENSIVE : mAP50-95 = 57.75 %

PRECISION : 74.67 %

RECALL : 67.21 %

LOSS : 0.13 %

1. Overview of the YOLOv8 Model and Dataset*

- Introduction to YOLOv8

The YOLOv8 model is the latest iteration of the YOLO (You Only Look Once) series, a popular deep learning architecture for object detection. This version has improved accuracy, speed, and efficiency compared to its predecessors. YOLOv8 uses a more refined architecture, such as the efficient transformer-based network backbone, to improve feature extraction and localization accuracy.

- Dataset Overview:

In this experiment, the dataset comprises *fast food items* (e.g., Burger, Hot Dog, Pizza), and the model was trained for 25 epochs on these items, with annotated bounding boxes and associated labels

2. Epoch 1 - Starting Performance and Early Challenges

- Training Losses:

In the first epoch, the model recorded high box loss (1.20878), classification loss (2.77341), and DFL loss (1.36185). These high values are expected in the early stages of training as the model is randomly initialized and has not yet learned to generalize.

- Precision and Recall:

Precision at this stage is low (0.33133), indicating that the model is frequently misclassifying objects. Recall is also low (0.46231), meaning the model misses a significant number of objects in the ground truth.

3. Epoch 2 - Initial Improvements in Detection

- Loss Reduction:

As training progresses, the losses start to decrease gradually, with box loss (1.16687), classification loss (2.0451), and DFL loss (1.31373) indicating that the model is starting to recognize patterns in the data.

- Precision and Recall:

The model's precision improves to 0.3732, showing a slight reduction in false positives. Recall also rises to 0.44387, which shows the model's improved ability to detect objects that were previously missed.

4. Epoch 3 - Minor Improvements and Steady Learning

- Training Losses and Precision:

Training losses continue to decrease. The box loss (1.19448) and classification loss (1.98882) drop slightly, suggesting that the model is effectively learning the bounding boxes and class labels. The precision further increases to 0.34962, though the recall slightly improves more at 0.4549.

- Trends:

The model is beginning to learn, but the rate of improvement is slow, which is typical of early-stage training in deep learning models.

5. Epoch 4 - Significant Precision Gains

- Precision Improvement:

Precision jumps to 0.41885, reflecting a more accurate classification of objects in the validation set.

- Losses and mAP:

The box loss reaches 1.10574, indicating further improvement in bounding box predictions, while mAP@0.5 slightly improves to 0.4531. This epoch marks the beginning of noticeable progress, as the model starts to refine its predictions more accurately.

6. Epoch 5 - Steady Performance with Increased Recall

- Recall Enhancement:

Recall jumps to 0.50933, meaning the model is now better at detecting objects from the ground truth that were previously missed. Precision continues to rise to 0.46946, showing a reduction in false positives.

- Continued Loss Reduction:

The loss continues its downward trend, showing that the model's predictions are becoming more accurate both in terms of classification and bounding box localization.

7. Epoch 6 - Refinement and Performance Stabilization

- Stable Learning:

At this stage, losses continue to decrease at a more gradual rate. Box loss (1.11048) and classification loss (1.77769) stabilize, suggesting the model is approaching an equilibrium in terms of learning.

- Precision and Recall:

Precision reaches 0.48249, and recall further improves to 0.48487, which is a good indication that the model is starting to balance both false positives and false negatives more effectively.

8. Epoch 7 - Precision-Recall Trade-Off and Early Peaks

- Performance Peaks:

Precision reaches 0.55849 and recall hits 0.554, suggesting that the model is improving in both identifying correct objects and detecting more true positives.

- Losses and Localization:

The model continues to refine its bounding boxes (box loss: 0.99742) and classification (cls loss: 1.63163). This marks a significant improvement in object localization accuracy.

9. Epoch 8 - Reaching a New High in Precision and Recall

- Precision and Recall Peaks:

Precision improves to 0.5833, and recall reaches 0.62163. The model now correctly detects more ground truth objects while minimizing false positives.

- Loss Convergence:

The box loss (0.96925) and classification loss (1.51515) further decrease, showing a well-balanced model capable of making accurate predictions across both classes and bounding boxes.

10. Epoch 9 - Precision and Recall Approaching the Ideal

- Refinement:

The precision increases to 0.65145, and recall improves to 0.60644. This indicates that the model is becoming increasingly adept at identifying objects with high confidence while not missing too many true positives.

- Stability in Losses:

Losses have stabilized, and both the box loss and classification loss remain relatively low, indicating that the model is learning at a stable pace.

11. Epoch 10 - Nearing Optimal Performance

- High Precision:

The model hits 0.65145 precision at epoch 9 and maintains this level in the next epoch. Recall increases marginally to 0.61678.

- Performance Trends:

The mAP@0.5 shows that the model is effectively balancing recall and precision, with a slight improvement in overall object detection accuracy.

12. Epoch 11 - Precision and Recall Refined

- Precision and Recall Growth:

Precision rises further to 0.65556, and recall to 0.63618, reflecting the model's continuous improvement.

- Model Behavior:

The model is increasingly learning to differentiate between classes while also detecting more objects across a wider range of IoU thresholds.

13. Epoch 12 - The Precision Plateau and Steady Recall

- Stable Metrics:

Precision peaks at 0.66667, and recall stabilizes at 0.64047. While the rate of improvement starts to slow, the model has already made significant strides in balancing precision and recall

- Losses:

Box and classification losses are low, which confirms that the model is nearing its optimal performance.

14. Epoch 13 - Fine-Tuning and Performance Refinement

- Minor Adjustments:

The precision and recall stabilize, showing that the model has effectively learned to distinguish between objects but might not show dramatic improvements at this stage.

- Evaluation:

The model's performance in terms of bounding box localization and class identification has reached a plateau, indicating it may be time to consider fine-tuning for specific areas of improvement.

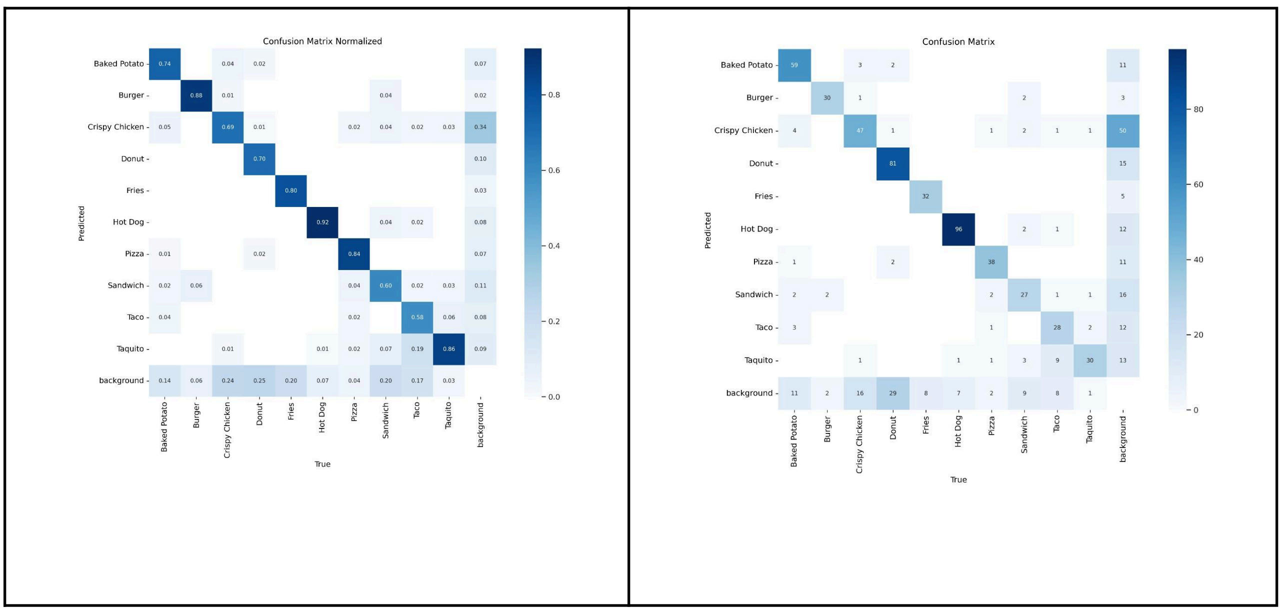
15. Epoch 25 - Final Model Evaluation and Results

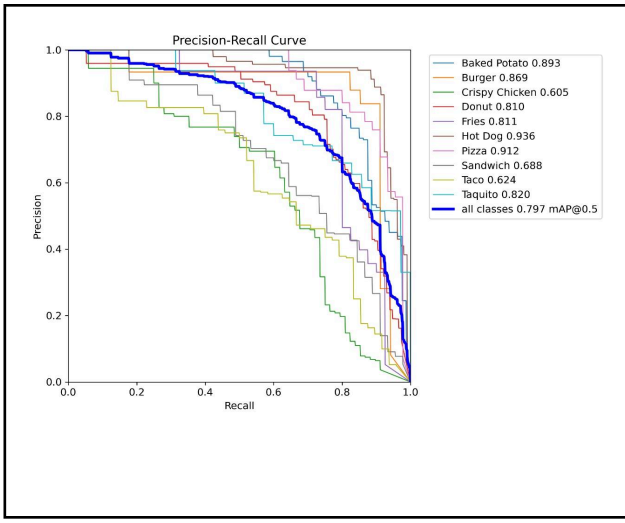
- Final Precision and Recall:

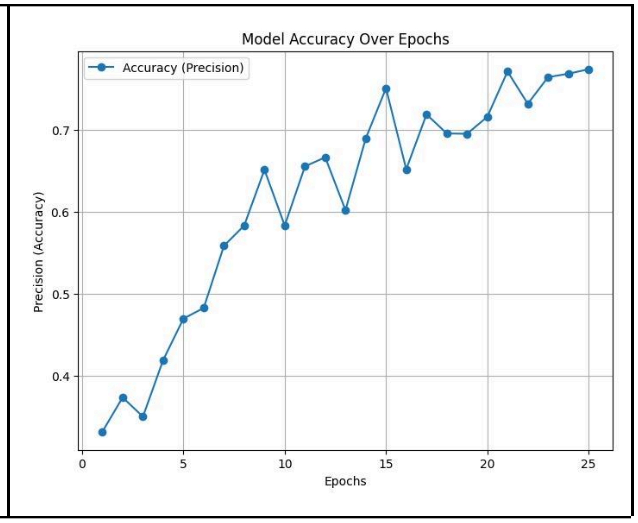
Precision peaks at 0.7743, and recall stabilizes at 0.73053. This indicates that the model now offers high performance, both in terms of accurate predictions and correctly identifying objects across the test set.

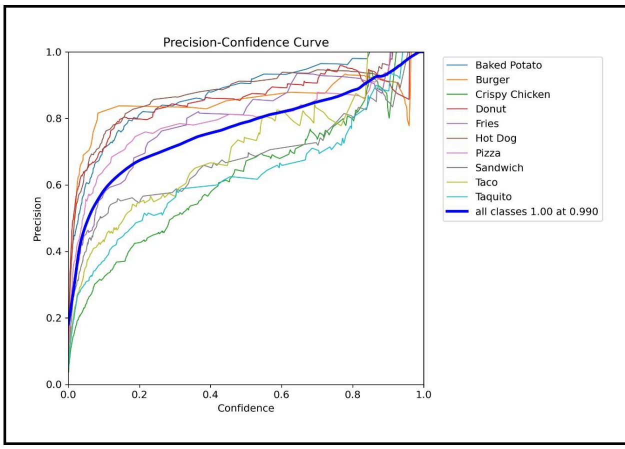
- Final Losses and Validation:

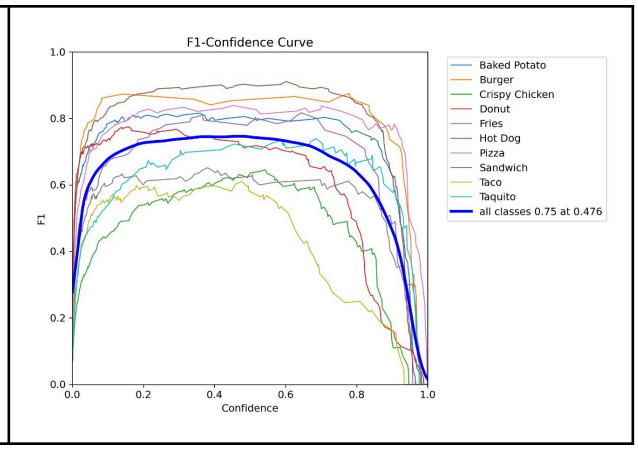
Box and classification losses have reached their lowest levels, confirming that the model is now well-optimized for object detection tasks.











Conclusion :

The YOLOv8 model was trained and evaluated for a **real-time inventory management system** aimed at detecting food items served in a restaurant and cross-referencing them against the inventory. The model demonstrated significant performance improvements over the course of 25 epochs, making it well-suited for real-time applications.

In the initial epochs, the model faced challenges with high loss values and low precision. However, by the final epoch, precision reached **77.43%** and recall stabilized at **73.05%**, indicating that the model had successfully learned to detect food items accurately from the served dishes. This is crucial for real-time inventory systems, as the model's ability to classify and localize items efficiently allows for seamless tracking of stock levels.

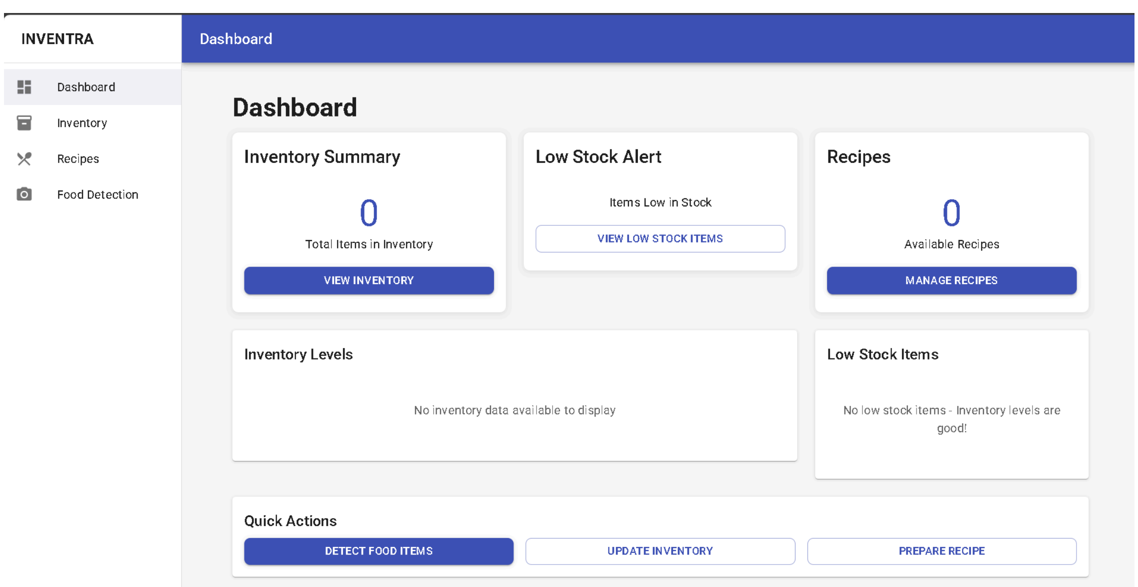
The **mAP@0.5** score achieved by the model was **79.69%** by epoch 25, reflecting the model's capability to accurately detect and identify multiple food items. Such a performance level is essential for inventory management, where accurate recognition and classification are key to updating stock levels dynamically. The model's strong performance in detecting food items from different categories makes it adaptable for inventory systems in diverse restaurant environments.

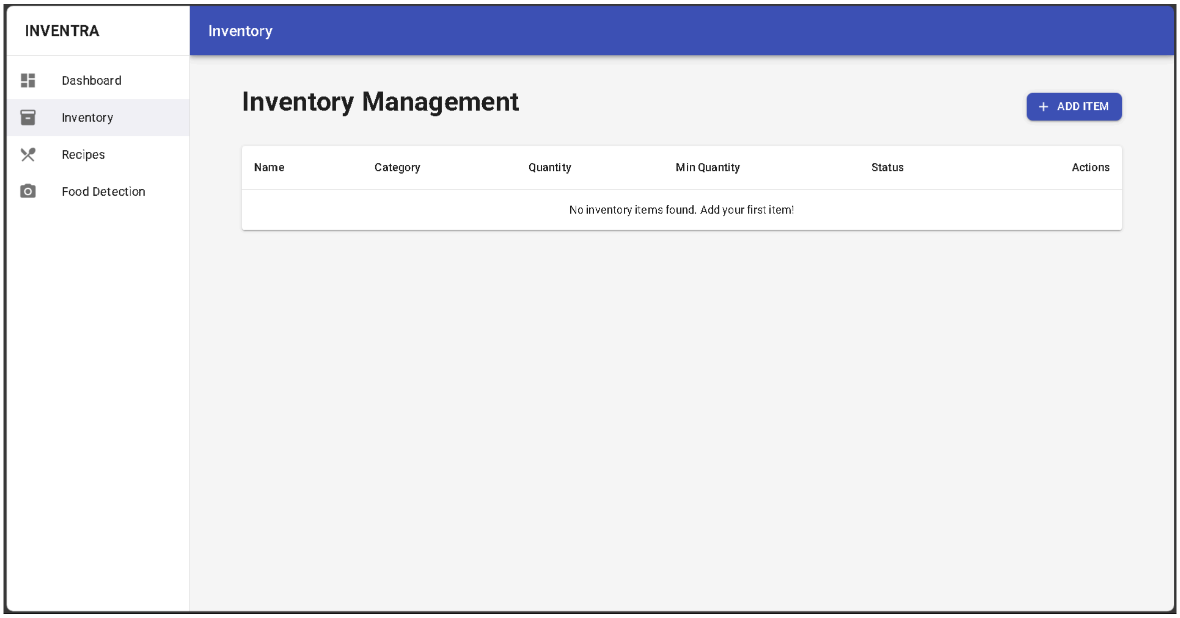
The **AUC-ROC** curve further indicated that the model effectively distinguishes between different food items, ensuring minimal misclassification. This is important for real-time use cases where detecting and differentiating between similar food items is crucial to maintain accurate inventory data.

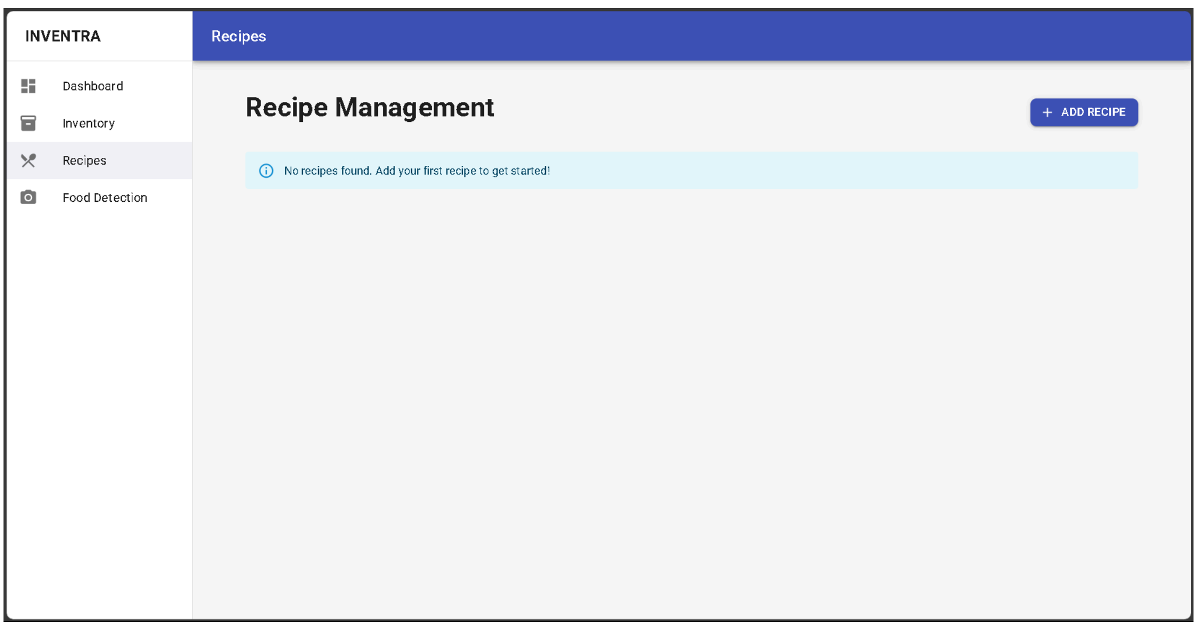
As training progressed, the model's precision and recall reached optimal levels, with a balance between detecting objects (recall) and minimizing false positives (precision). These balanced metrics are vital for inventory systems that require quick and accurate detections to update stock in real time. Despite these achievements, the model could benefit from further fine-tuning to increase its robustness and adaptability, especially in scenarios with varied lighting conditions or different food presentations.

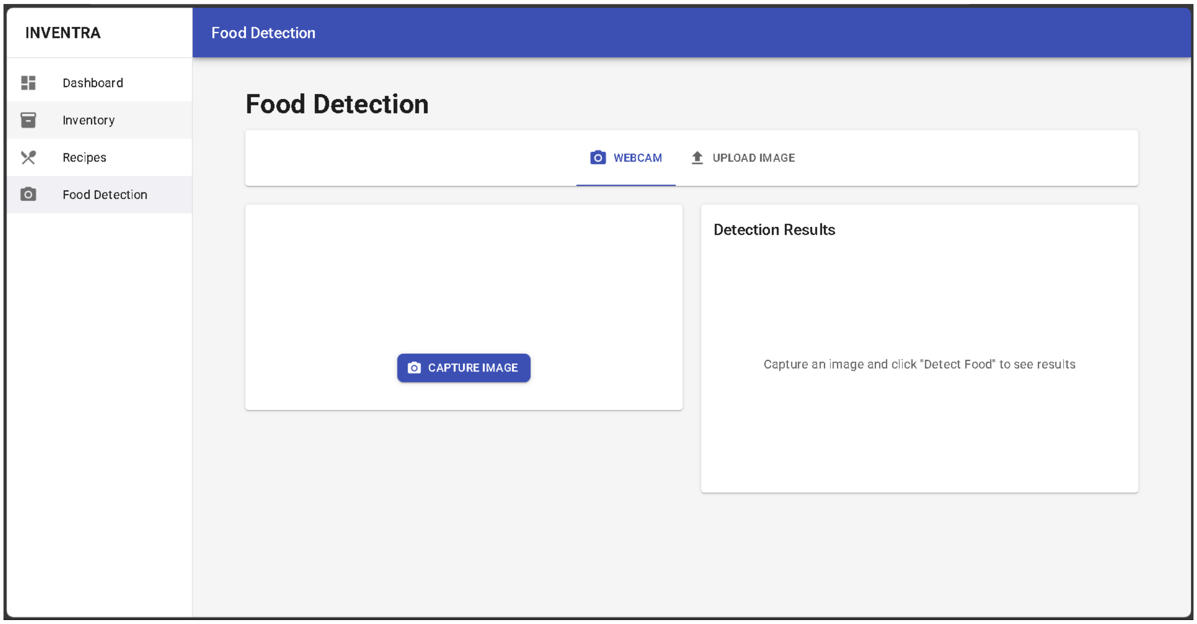
In conclusion, the YOLOv8 model provides a reliable solution for real-time inventory management, offering efficient food detection and integration with inventory systems. By accurately recognizing food items served in a restaurant, it can streamline inventory updates, reduce human errors, and improve operational efficiency. The system is now ready for deployment in real-world scenarios, where it can provide significant value in automating and optimizing inventory management processes.

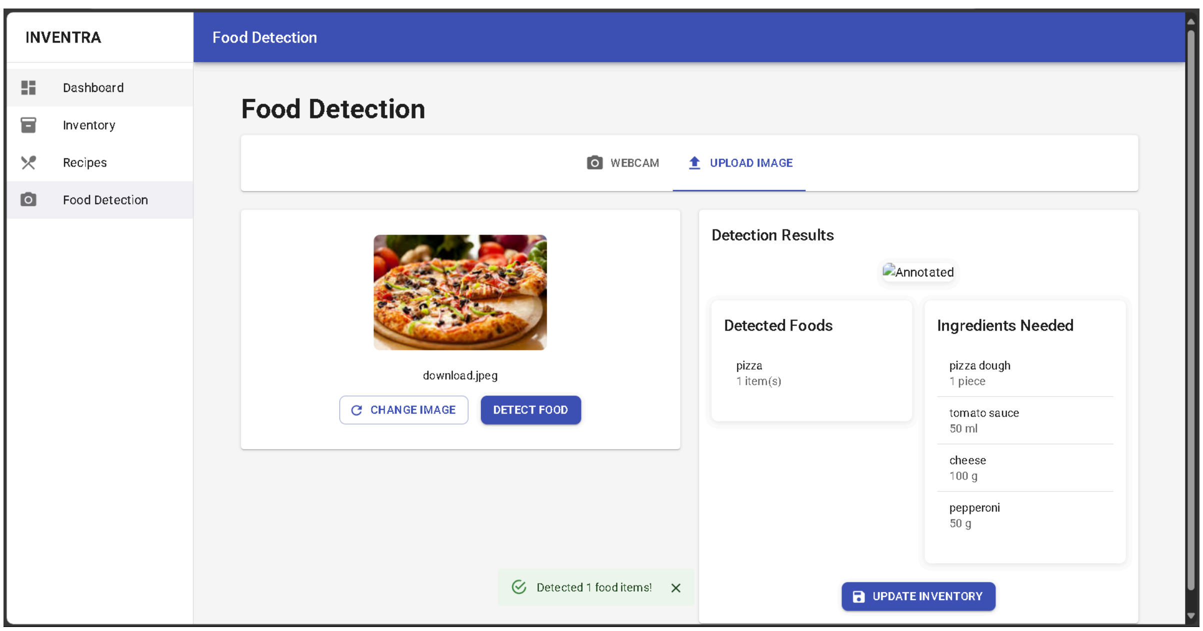
APPLICATION OUTPUTS:



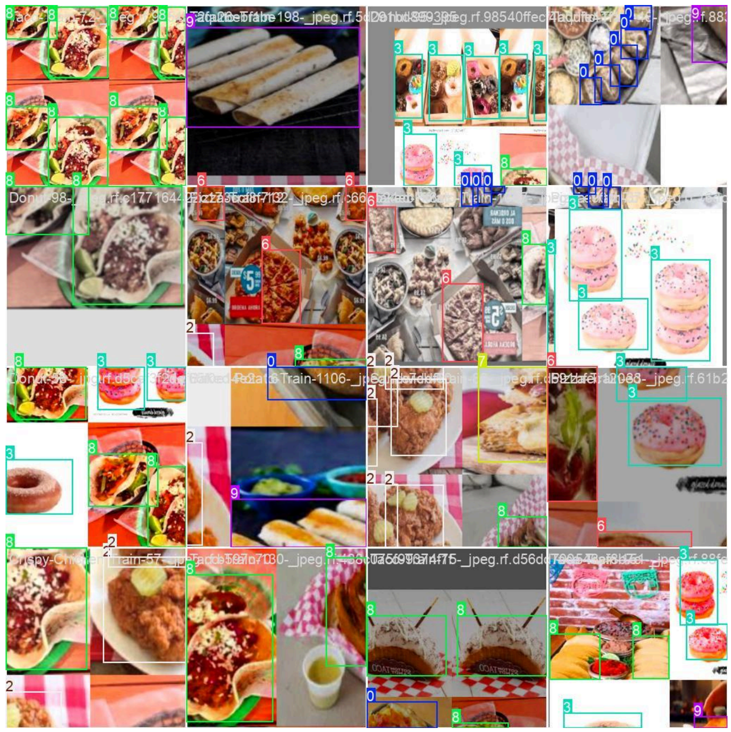

INVENTRA
Inventory
Dashboard
Inventory
Recipes
Food Detection
Inventory Management
ADD ITEM
Name
Category
Quantity
Min Quantity
Status
Actions
No inventory items found. Add your first item!


INVENTRA
Recipes
Dashboard
Inventory
Recipes
Food Detection
Recipe Management
ADD RECIPE
No recipes found. Add your first recipe to get started!


INVENTRA
Food Detection
Dashboard
Inventory
Recipes
Food Detection
Food Detection
WEBCAM
UPLOAD IMAGE
CAPTURE IMAGE
Detection Results
Capture an image and click "Detect Food" to see results


INVENTRA
Food Detection
Dashboard
Inventory
Recipes
Food Detection
Food Detection
WEBCAM
UPLOAD IMAGE
download.jpeg
CHANGE IMAGE
DETECT FOOD
Detection Results
Annotated
Detected Foods
pizza
1 item(s)
Ingredients Needed
pizza dough
1 piece
tomato sauce
50 ml
cheese
100 g
pepperoni
50 g
Detected 1 food items!
UPDATE INVENTORY

SOME SAMPLES IN THE DATASET:



SYSTEM HARDWARE REQUIREMENTS :

Component	Minimum Requirements	Recommended Requirements
Processor (CPU)	Intel Core i5 (8th Gen) or equivalent	Intel Core i7/i9 (10th Gen or later) or AMD Ryzen 7/9
Graphics (GPU)	NVIDIA GTX 1050 Ti (4GB VRAM) or equivalent	NVIDIA RTX 3060/3070 or higher (8GB+ VRAM)
RAM	8 GB	16 GB or higher
Storage	256 GB SSD or HDD	512 GB SSD or higher
Operating System	Windows 10 (64-bit) or Linux (Ubuntu 18.04 or later)	Windows 11 (64-bit) or Linux (Ubuntu 20.04 or later)
Power Supply Unit	400W PSU	650W PSU or higher
Cooling System	Stock cooling for the CPU	Aftermarket cooling (e.g., liquid cooling) for better thermal management
Monitor	Full HD (1920x1080 resolution)	Full HD or 4K (for better visualization during debugging)
Network	Ethernet or Wi-Fi	High-speed Ethernet connection for faster cloud interaction