**Literature Survey: Real-Time Food Object Detection and Classification for Smart Kitchen Assistants**

**1. Introduction** With growing interest in personal health, dietary monitoring, and smart kitchen technologies, AI-based food recognition systems have emerged as powerful tools. Real-time food object detection and classification has the potential to revolutionize food logging, nutrition tracking, and kitchen automation. This survey details a practical implementation of such a system, encompassing dataset preparation, model selection, annotation methodology, system deployment, and user interface development.

**2. Dataset Acquisition and Preprocessing** The foundation of this project began with the acquisition of a food image dataset from Kaggle. The preprocessing pipeline included: - **Normalization** to standardize image data, - **Class balancing** to reduce label skew and improve fairness during training, - **Data augmentation** to expand the dataset's diversity and robustness.

The dataset was structured in COCO format to ensure compatibility with contemporary object detection frameworks.

**3. Initial Model Selection: RF-DETR** The initial model selected for experimentation was **RF-DETR (Region-Focused Detection Transformer)**, a state-of-the-art vision transformer architecture provided via Roboflow. However, practical issues arose: - **Lack of MPS (Metal Performance Shaders) support** limited its GPU compatibility on Apple hardware. - **High computational demand** rendered it impractical for local training without specialized hardware.

Despite robust documentation and support, RF-DETR was found unsuitable for the resource constraints of the local development environment.

**4. Transition to YOLOv11n** After identifying limitations with RF-DETR, the project pivoted to **YOLOv11n**, a lightweight object detection model with: - Low latency and efficient inference, - Compatibility with COCO-style datasets, - Acceptable accuracy benchmarks for real-time performance.

This shift required reformatting the dataset to adhere to YOLOv11n's expected input structure, but provided significant benefits in computational efficiency.

**5. Automated Annotation using Grounding DINO** To annotate images programmatically, **Grounding DINO**, a zero-shot object detection model, was employed. This tool used class-specific text prompts to generate bounding boxes.

The annotation script accomplished the following: - Inferred bounding boxes for each image using class prompts, - Converted annotations to both YOLO and COCO formats, - Discarded unannotated images, - Rebalanced and partitioned the dataset into training, validation, and test sets.

This step dramatically reduced manual labeling effort while maintaining sufficient annotation quality.

**6. Training Challenges and Cloud-Based Solution** Initial attempts to train the YOLOv11n model on a MacBook using MPS resulted in severe memory leaks. Specifically, RAM usage doubled after each training epoch, leading to a slowdown and eventual instability.

To overcome this, the training pipeline was migrated to **Google Colab**, which provided: - Access to GPU acceleration, - A stable and reproducible environment, - Support for the **Ultralytics YOLO** framework.

The model was trained for 10 epochs using the custom dataset, yielding the following results: - **mAP50**: 0.813, - **mAP50-95**: 0.769.

High-performing categories included Besan Cheela, Jalebi, and Poha, while moderately performing classes included Idli and Dahl.

**7. Inference and Real-Time Testing** Once trained, the model was evaluated using static images via OpenCV. The inference script demonstrated reliable real-time object detection performance with low latency.

**8. User Interface Development with Streamlit** A user interface was created using **Streamlit**, providing multiple interaction modes: - Real-time webcam-based food detection, - Image upload and snapshot support, - Annotated image display with overlaid bounding boxes.

The interface was extended to include a nutritional insights panel that: - Displayed macronutrients and micronutrients for each detected food item, - Showed estimated per-item and total caloric intake, - Linked to recipes and ingredient lists.

**9. Nutrition Tracking and Logging** To provide comprehensive dietary monitoring, the system incorporated: - A nutrition CSV mapped to food class names, - Real-time calorie computation based on detection counts, - Persistent logging of each detection session, - Visualization of daily and historical caloric trends using Plotly, - Export functionality for food logs and nutrition summaries.

**10. Technical Architecture Overview** - **Backend**: PyTorch, OpenCV, Grounding DINO, Ultralytics YOLO - **Frontend**: Streamlit, Streamlit-WebRTC, Plotly - **Data Processing**: Pandas, PIL, JSON, COCO/YOLO converters - **Training Infrastructure**: Google Colab

**11. Challenges Encountered** - **Model Incompatibility**: RF-DETR was incompatible with macOS-based GPU acceleration. - **Annotation Gaps**: Some classes were inconsistently annotated and required removal or manual correction. - **Memory Leaks**: MPS-based training proved inefficient, necessitating a shift to cloud compute.

**12. Future Directions** To improve the system's accuracy and applicability: - **Depth estimation** could be added to approximate food volume and improve calorie estimation. - **Instance segmentation** may enhance boundary accuracy for overlapping food items. - **Edge optimization** is necessary for deployment on low-power devices.

These enhancements would offer more precise nutritional insights but may increase computational overhead.

**13. Conclusion** This project demonstrates a viable and scalable pipeline for real-time food detection and nutritional analysis. By combining lightweight object detection models, automated annotation, and an intuitive interface, it provides a strong foundation for applications in health monitoring and kitchen automation. Despite computational challenges, the adopted methods ensured real-time usability and formed a platform for future development in AI-driven dietary assistance systems.