System Software

Input layer: Camera , Mic , Sensors

Processing layer:

Vision Models:

YOLO: is a family of computer vision models that are widely used for real-time object detection**.** YOLO's speed is a major advantage, allowing it to process images and video frames very quickly

How YOLO Works :

Image Division: The input image is divided into a grid.

Prediction: Each grid cell predicts bounding boxes, confidence scores for those boxes, and class probabilities for potential objects within the cell.

Filtering: Non-maximum suppression is used to filter out redundant bounding boxes, keeping only the most confident predictions.

The YOLO family has seen significant advancements since its inception, with various versions (YOLOv1, YOLOv2, YOLOv3, etc.) introducing improvements in accuracy, speed, and functionality. More recent developments like YOLO-World and YOLO-E have further expanded YOLO's capabilities to include zero-shot and open-vocabulary object detection, enabling the detection of objects based on text descriptions or example images.

OCR: (OCR) allows text in images to be understandable by machines, allowing programs and scripts to process the text. OCR is commonly seen across a wide range of applications, but primarily in document-related scenarios, including document digitization and receipt processing.

Tesseract + PyTesseract with Open CV improving photo's

EasyOCR

PaddleOCR

Vision Models Freshness: Computer vision technique for freshness estimation from segmented eye of fish image.

----

vision models detection : AI systems that analyze images and videos to identify and locate objects within them.

NLP Models:

1-STT

is an essential component for **creating voice-powered experiences** that delight users. A subset of automatic speech recognition (ASR), STT algorithms enable you to apply text-based natural language processing (NLP) techniques to a user's intentions. This makes speech-to-text perfect for use cases like:

- generating video captions
- transcribing meetings
- converting voice to plain text for analysis

2-TTS

The primary function of TTS technology is to convert written text into spoken words. NLP plays a crucial role in this process by analyzing the text and generating a human-like voice that accurately reflects the meaning and tone of the text

----

3-Feedback Correction

Auto text correction is a fundamental tool in modern Natural Language Processing (NLP) that upgrades user involvement by decreasing typographical and linguistic errors in composed content. This extends leverages progressed NLP procedures to create a strong and proficient auto-correction framework that precisely predicts and corrects content input errors in real-time. The framework utilizes a combination of deep learning models, such as sequence-to-sequence models and transformers, in conjunction with conventional strategies like n-grams and edit distance algorithms. By analyzing the context and structure of input sentences, the model is able to offer relevant rectifications, making it versatile to different dialects and user composing styles. Moreover, we investigate diverse assessment measurements and compare different calculations to optimize the precision and productivity of the correction system. The arrangement has potential applications in word processors, mobile keyboards, and other content input platforms where high accuracy and low latency are crucial.

Database Shopping List

A database-driven shopping list application would typically involve several tables to store and manage the data effectively. Here's a breakdown of the key tables and their relationships

1. List Table:

This table stores information about each individual shopping list.

```
CREATE TABLE Lists (
   list_id INT PRIMARY KEY AUTO_INCREMENT,
   list_name VARCHAR(255) NOT NULL,
   creation_date DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

2. `Items` Table:

This table contains details about the items available for purchase. It can serve as a master list of common grocery or household items.

```
CREATE TABLE Items (
    item_id INT PRIMARY KEY AUTO_INCREMENT,
    item_name VARCHAR(255) NOT NULL UNIQUE,
    category VARCHAR(100), -- e.g., Produce, Dairy, Cleaning Supplies
    unit_of_measure VARCHAR(50) -- e.g., kg, dozen, bottle
);
```

Database User Preferences

Methods for Storing User Preferences in Databases:

- **Relational Database Tables:**
- **Dedicated Preferences Table:** A common approach involves creating a separate table to store user preferences. This table typically includes columns for a user_id, a preference_key (e.g., 'theme', 'notification_type'), and a preference_value (e.g., 'dark', 'email'). This offers flexibility for adding new preferences without altering the table schema.

JSON Storage: Modern relational databases often support storing JSON data types. User preferences can be stored as a JSON object within a single column in a user table, providing a flexible, schema-less approach.

Crowd Density Detection

uses computer vision and deep learning to analyze images and videos, primarily from surveillance cameras, to estimate the number of people in an area and identify potential overcrowding or safety risks in real-time. Key technologies involve Convolutional Neural Networks (CNNs) and models like YOLO, which are trained on datasets to accurately identify and count people or predict crowd density maps, enabling proactive management of public spaces like venues, stations, and malls.

1. Input:

The system receives image or video feeds, often from real-time CCTV cameras.

- **Crowd Counting Models:**

  Specialized models, such as MCNN (Multi-column CNN), are designed to capture features at different scales to estimate crowd counts or densities.

- **Forgotten Item Reminder**

Database SQLite/local list: SQLite is a lightweight, serverless, and self-contained relational database management system that stores data in a local file. It is commonly used in applications that require a local, embedded database, such as mobile apps, desktop applications, and embedded systems.

Key aspects of using SQLite for a local list:

- **Database Creation and Connection:**

  SQLite databases are typically created as a file on the local file system.

  Applications connect to this file to interact with the database.

- **Schema Definition:**

  Tables are defined within the database to structure the data.

  For a "local list," a table would be created to store the list items, including columns for relevant attributes (e.g., item name, description, status).

- **Data Manipulation:**

  Insertion: New list items are added to the table using SQL INSERT statements.

  Retrieval: List items are fetched using SQL SELECT statements, often with WHERE clauses for filtering or ORDER BY for sorting.

  Updating: Existing list items are modified using SQL UPDATE statements.

  Deletion: List items are removed using SQL DELETE statements.

- **Integration with Programming Languages/Frameworks:**

  SQLite can be integrated into various programming environments using specific libraries or frameworks (e.g., sqlite3 module in Python, SQLite.NET in .NET MAUI, Room persistence library in Android).

  These tools provide an interface to interact with the SQLite database from within the application's code.
- **Local Persistence:**
  Data stored in a SQLite database persists across application sessions, meaning the list items are saved even after the application is closed and reopened.

  - **Function : Compares current purchases with the entered list.**
  - **Output: My voice , 'I haven't bought the oil yet'**

User Feedback Correction:

NLP Speech-to-Text:

Correction loop→ The user says "No that's not it" the system records the feedback and corrects.
Seeing AI: Dynamic interaction not found in adventage.

Freshness Detection:
Freshness detection involves using sensory evaluation or technology like gas sensors, spectroscopy, mass spectrometry, and AI-powered image analysis to assess food quality and spoilage. While traditional human evaluation is common, its subjectivity leads to increased use of automated, objective methods to identify changes in smell, color, texture, or other indicators of staleness for fruits, vegetables, and fish, ensuring consumer safety and reducing food waste.

- **Technological Approaches**

Artificial Intelligence (AI) and Computer Vision:

Deep Learning: Models like Convolutional Neural Networks (CNNs) analyze images of fruits and vegetables to identify signs of spoilage.

Robotics: Automated systems can use cameras and AI to identify and separate rotten produce on a conveyor belt.

Health App Integration:

Health application integration is the process of connecting disparate health systems, devices, and software applications to facilitate secure data exchange and interoperability, improving patient care, operational efficiency, and user experience. This is achieved using Application Programming Interfaces (APIs), which act as communication protocols, and often adhere to standards like FHIR (Fast Healthcare Interoperability Resources) for seamless data sharing. Key integrations include connecting Electronic Health Records (EHRs), wearable devices, pharmacy management systems, and platforms like Google's Health Connect for a holistic view of patient health.

Google Fit/Apple Health .apply with API connection

Checks food against dietary restrictions (مثلاً السكر/الضغط).

Quick Pick Mode:

Database of frequently bought items.

Path optimization inside store (integrated with SLAM + Google Maps API)

Offline Functionality

YOLOv10-Nano + OCR models run locally on Raspberry Pi.

Strength vs Seeing AI: full features even with no internet .

Technology Stack (Software-Focused):

Computer Vision: OpenCV, YOLOv10-Nano, Mask R-CNN.

OCR: EasyOCR (Arabic), Tesseract (English).

Speech: Whisper/Vosk (STT), Tacotron2 (TTS Arabic).

Database: SQLite (local shopping lists, maps).

Navigation: ORB-SLAM3 + Google Maps API.

Integration: Flutter (companion app), Firebase (optional caregiver sync).

Software Development Phases

- **Phase 1**:
  - Product recognition, OCR, simple TTS.
- **Phase 2**:
  - Add crowd density detection + forgotten item reminder.
- **Phase 3**:
  - Feedback correction, freshness detection, quick pick.
- **Phase 4**:
  - Health integration + advanced personalization.
