Project Title	**Fruit Object Detection using Deep Learning and application using Streamlit and Hosted on AWS**
Skills take away From This Project	**Image data preprocessing and augmentation** **Object detection using Convolutional Neural Networks (CNNs)** **Transfer Learning with pre-trained models (YOLOv8 / Faster R-CNN / SSD)** **Model evaluation using precision, recall, and mAP** **Visualization of bounding boxes and class predictions** **Deployment-ready model preparation**
Domain	**Manufacturing Quality Check, Computer Vision – Object Detection**

Problem Statement:

Identify and localize fruits (banana, orange, and apple) within an image by drawing bounding boxes around them and labeling each detected fruit correctly. The model should work efficiently on unseen images and maintain accuracy across different lighting and orientation conditions.

Business Use Cases:

Smart Retail: Automate fruit recognition and counting for billing or stock monitoring.
Agriculture: Identify fruits on trees for yield estimation.
Food Industry: Automate fruit sorting on conveyor belts using real-time camera feeds.
Health Tech: Assist calorie-tracking apps by identifying fruits from meal images.

Approach:

Data Collection:

1. Use the provided dataset containing 240 images for training and 60 images for testing, each with images of bananas, oranges, and apples.

Data Annotation:

1. Annotate images with bounding boxes for each fruit using tools like LabelImg or Roboflow to generate YOLO-format or Pascal VOC XML annotations.

Data Preprocessing:

1. Resize images to a standard input size (e.g., 416x416 or 640x640).
2. Apply augmentation: random rotation, brightness, flipping, and scaling to improve robustness.

Model Training:

1. Use YOLOv8 or Faster R-CNN pre-trained on COCO dataset.
2. Fine-tune the model on your fruit dataset for transfer learning.
3. Split data: 80% train, 20% validation.

Model Evaluation:

1. Evaluate performance on test data using metrics such as Precision, Recall, F1 Score, and mean Average Precision (mAP).

Visualization:

1. Plot detected bounding boxes on test images to visually confirm performance.

Deployment:

1. Export the trained model and use it in a Streamlit app or Flask API for live detection.

5. ***Feel free to also explore any other Models for the dataset which has been attached.***

Results:

1. The model successfully detects and classifies banana, orange, and apple with high confidence.
2. Achieved mAP@0.5 = 0.93 and F1-score = 0.90 on the test dataset.

Project Evaluation metrics:

Project Evaluation metrics:

Primary Metrics (Robust to Class Imbalance):

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): For binary classification of normal vs. anomaly.

Area Under the Precision-Recall Curve (AUC-PR): Especially critical for highly imbalanced datasets where anomalies are rare.

Secondary Metrics (Threshold-Dependent):

F1-Score: For the anomaly class, at an optimized threshold.

Precision, Recall, Specificity: At various operating points.

Localization Metrics (if pixel-level masks are available):

Per-pixel AUC-ROC: To evaluate the accuracy of anomaly maps.

IoU (Intersection over Union): Between predicted anomaly masks and ground truth.

Inference Speed/Latency: Crucial for real-time industrial deployment.

Robustness: To variations in lighting, background, and minor normal object variations.

Technical Tags:

DeepLearning, ObjectDetection, ComputerVision, YOLOv8, FasterRCNN
ImageProcessing, TransferLearning, TensorFlow, PyTorch

Data Set:

Dataset Link: **Data**

Data Set Explanation:

1. Train Folder: 240 images divided equally among classes (banana, orange, apple).
2. Test Folder: 60 images (20 per class).
3. Each image contains one or more fruits, possibly with occlusions or different lighting.
4. Annotations: Each image has bounding box coordinates specifying fruit positions and labels.

Project Deliverables:

1. Jupyter Notebook / Python Script with end-to-end implementation
2. Trained Model File (best.pt or model_final.pth)
3. Evaluation Report with metrics and confusion matrix
4. Visualization Results (detected fruits with bounding boxes)
5. ReadMe Documentation
6. Streamlit App or Flask API for live fruit detection

MANDATORY: Application hosted on AWS or GCP for detecting the Anomalies from the user input

Project Guidelines:

Resize images to consistent dimensions (e.g., 640x640).

Normalize pixel values between 0–1.

Augment data using techniques like:

- Horizontal/vertical flips
- Random rotations
- Brightness and contrast variations
- Gaussian noise

Timeline:

Define the project timeline, including milestones and deadlines.

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday-Saturday (4:00PM to 5:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

For DE/BADM project/class topic doubt slot clarification session:

Booking link : **https://forms.gle/NtkQ4UV9cBV7Ac3C8**

Session timing:

For DE: 04:00 pm to 5:00 pm every saturday
For BADM 05:00 to 07:00 pm every saturday

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open only on Saturday (after 2 PM) and Sunday on Every Week

Timing:

For BADM and DE
Monday-Saturday (11:30AM to 1:00PM)

For DS and AIML
Monday-Saturday (05:30PM to 07:00PM)

Booking link : **https://forms.gle/1m2Gsro41fLtZurRA**