**Fruit Object Detection using YOLOv8 with Streamlit Application Hosted on AWS**

**Author:** Gowsalya S.

**Program:** Data Science

**Domain:** Computer Vision – Object Detection

## Abstract

Object detection plays a vital role in modern computer vision applications such as smart agriculture, retail automation, and quality inspection. This paper presents a fruit object detection system capable of identifying and localizing Apples, Bananas, and Oranges using the YOLOv8 deep learning model. A custom dataset was annotated in YOLO format and trained using the Ultralytics framework on a GPU-enabled environment. Data augmentation techniques were applied to enhance model generalization.

The trained model was deployed as a web-based application using Streamlit and hosted on Amazon Web Services (AWS), enabling real-time fruit detection through user-uploaded images. Experimental results demonstrate high accuracy with strong precision, recall, and mean Average Precision (mAP), validating the effectiveness of the proposed approach for real-world deployment.

## Keywords

Object Detection, YOLOv8, Deep Learning, Fruit Detection, Computer Vision, Streamlit, AWS Deployment

## 1. Introduction

Object detection is a fundamental task in computer vision that involves identifying objects and their locations within an image. Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have significantly improved detection accuracy and inference speed.

YOLO (You Only Look Once) is a single-stage object detection model that performs detection in real time. YOLOv8, the latest version from Ultralytics, provides improved accuracy, faster inference, and a flexible deployment pipeline.

This project focuses on detecting three fruit categories—Apple, Banana, and Orange—using YOLOv8. The system is designed to work efficiently on unseen images and is deployed as a cloud-based Streamlit application hosted on AWS.

## 2. Dataset Description

### 2.1 Class Labels

* Apple (Class 0)
* Banana (Class 1)
* Orange (Class 2)

### 2.2 Dataset Split

* Training Set: 240 images
* Validation Set: Included during training
* Test Set: 60 images (20 per class)

Each image may contain one or more fruits under different lighting conditions, orientations, and partial occlusions.

### 2.3 Project Structure

project/  
├── yolo\_data/  
│ ├── images/  
│ │ ├── train/  
│ │ ├── valid/  
│ │ └── test/  
│ ├── labels/  
│ │ ├── train/  
│ │ ├── valid/  
│ │ └── test/  
├── data.yaml  
├── best.pt  
├── fruits\_streamlit.py  
└── README.md

Annotations were created in YOLO format using normalized bounding box coordinates.

## 3. Methodology

### 3.1 Model Architecture

The YOLOv8s model from the Ultralytics framework was selected due to its balance between speed and accuracy. The model predicts bounding boxes and class probabilities in a single forward pass, making it suitable for real-time applications.

### 3.2 Data Augmentation

To improve robustness and reduce overfitting, the following augmentation techniques were applied:

* Horizontal Flip
* Brightness and Contrast Adjustment
* Rotation
* Color Jitter
* Motion Blur and Gaussian Blur
* Random Shadows

### 3.3 Training Configuration

* Model: YOLOv8s
* Epochs: 50
* Image Size: 640 × 640
* Batch Size: 8
* Optimizer: Default YOLOv8 optimizer
* Environment: Google Colab (GPU-enabled)
* Early Stopping: Enabled

## 4. Experimental Results

### 4.1 Per-Class Performance

| Class | Precision | Recall | F1-Score |
| --- | --- | --- | --- |
| Apple | 0.913 | 0.9999 | 0.995 |
| Banana | 0.845 | 0.9990 | 0.995 |
| Orange | 0.951 | 0.9990 | 0.993 |

### 4.2 Overall Performance Metrics

| Metric | Value |
| --- | --- |
| mAP@0.5 | 0.9105 |
| mAP@0.5–0.95 | 0.6671 |

### 4.3 Confusion Matrix

[28 0 3 4  
 0 26 0 11  
 0 0 20 4  
 0 14 0 0]

## 5. Deployment Using Streamlit and AWS

### 5.1 Streamlit Application

The trained model (best.pt) was integrated into a Streamlit application that allows users to upload images and view detected fruits with bounding boxes and labels.

Key features:

* Image upload interface
* Real-time YOLOv8 inference
* Bounding box visualization
* Lightweight and user-friendly UI



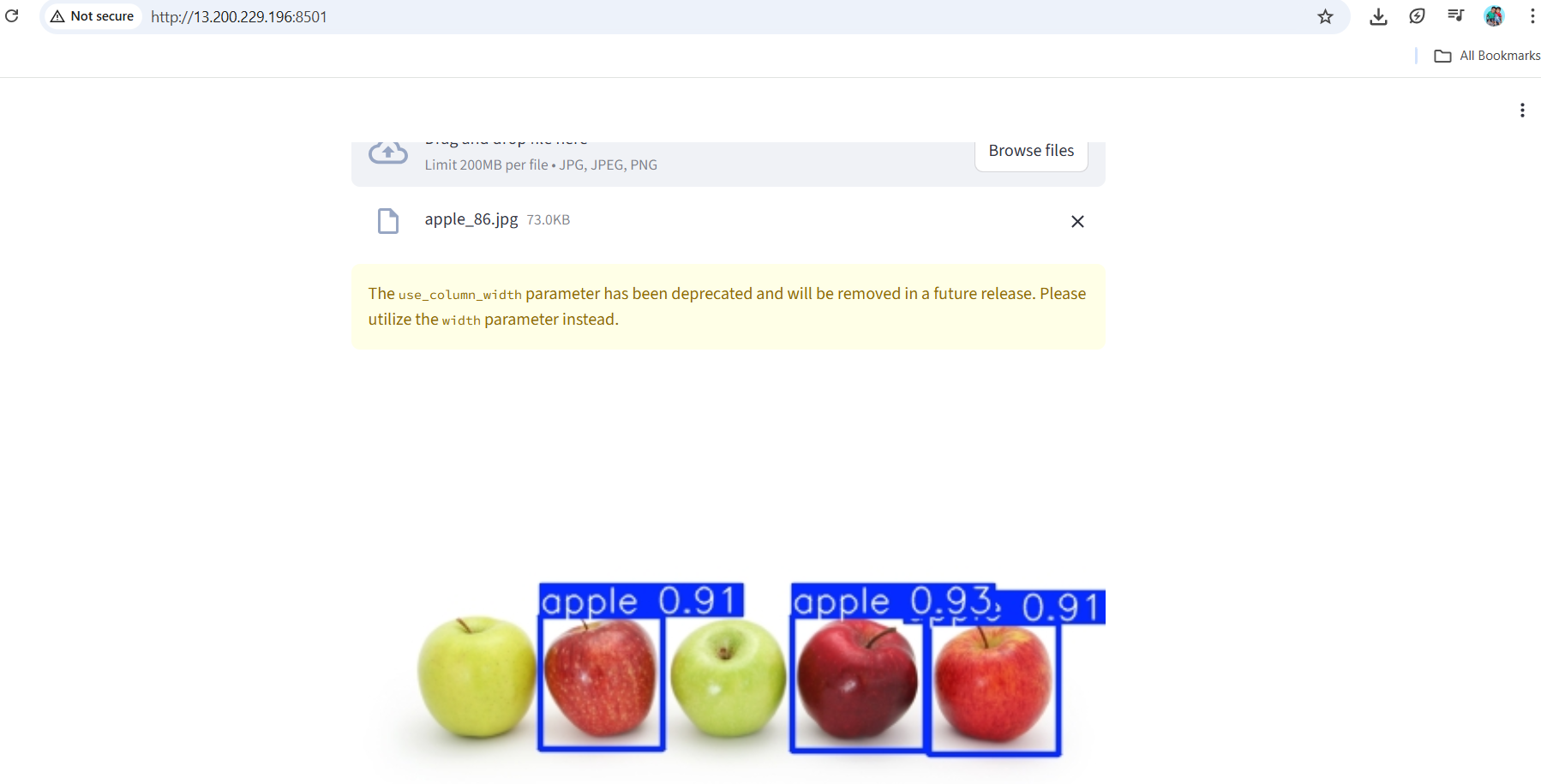
### 5.2 AWS Hosting

The Streamlit application was deployed on an **AWS EC2 Ubuntu instance** with the following setup:

* Instance Type: CPU-based EC2
* Python Virtual Environment
* Required libraries: PyTorch (CPU), Ultralytics, OpenCV, Streamlit
* Model loading using absolute file paths
* Public access via EC2 public IP and port 8501

Deployment enables real-time inference through a web browser.

External URL : <http://13.200.229.196:8501>



## 6. Business Use Cases

* **Smart Retail:** Automated fruit recognition and counting for billing and inventory monitoring
* **Agriculture:** Fruit detection on trees for yield estimation
* **Food Industry:** Automated fruit sorting on conveyor belts
* **Health Tech:** Fruit identification for calorie tracking applications

## 7. Conclusion

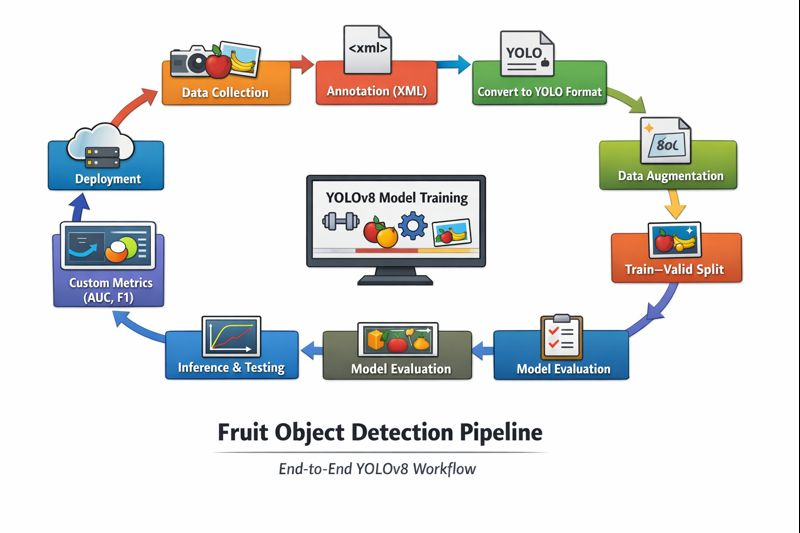
The project successfully implemented a YOLOv8-based fruit object detection system capable of accurately detecting apples, bananas, and oranges. The model achieved high precision, recall, and mAP scores, demonstrating strong generalization performance.

Integration into a Streamlit application and deployment on AWS highlights real-world applicability. Future enhancements may include expanding the dataset, adding more fruit classes, and enabling real-time video detection.

## Reference files :

## 

## Pipeline:



## References

[1] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” *CVPR*, 2016. [2] Ultralytics, “YOLOv8 Documentation,” Available: https://docs.ultralytics.com [3] A. Bochkovskiy et al., “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv*, 2020.