Project Documentation: Bird Counting and Weight Estimation (CCTV Video)

Abstract

This project presents a computer vision–based system for **Bird Counting and Weight Estimation** using fixed-camera poultry CCTV videos. The system leverages deep learning–based object detection, multi-object tracking, and API deployment to automatically detect birds, track them across frames, estimate population count over time, and infer relative weight using visual cues. The solution is deployed using a FastAPI backend and produces annotated video outputs along with structured JSON responses.

Objective

Detect The primary objectives of this project are:

- To detect birds accurately from poultry farm video footage.
- To count birds over time using stable object tracking.
- To estimate bird weight using a visual proxy (bounding box area).
- To expose results via a RESTful API.
- To generate annotated videos for visual verification.

1. Dataset

- **Source**: Poultry CCTV videos (Kaggle / provided sample videos)
- **Type**: Video data captured using a fixed overhead camera
- **Content**: Multiple birds moving freely within a poultry shed
- **Format**: `.mp4`
- **Assumption**: Single camera, constant camera angle, no sudden lighting changes

Due to limited access to real-time farm data, publicly available poultry datasets were used for prototyping.

2. Data Preprocessing & Augmentation

Preprocessing:

- Video frames extracted using OpenCV.
- Frames resized for faster inference.
- RGB color normalization applied.
- Frame-by-frame processing to maintain temporal consistency.

Augmentation (Conceptual):

- Horizontal flipping
- Brightness and contrast adjustment
- Scaling and cropping (for training robustness)

Augmentation improves generalization when fine-tuning detection models.

3. Model & Framework

- **Object Detection Model**: YOLO (You Only Look Once)
- **Tracking Method**: IOU-based multi-object tracker
- **Backend Framework**: FastAPI
- **Libraries Used**:
- OpenCV
- NumPy
- Ultralytics YOLO
- FastAPI
- Uvicorn

YOLO is chosen for its real-time detection capability and high accuracy.

4. Training Setup & Hyperparameters

(Pretrained model used – no full training due to time and data constraints)

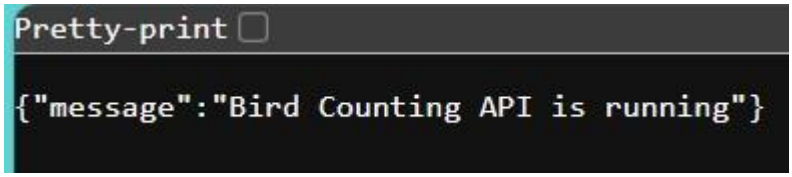
- Model: YOLO pretrained weights
- Confidence threshold: 0.4
- IOU threshold: 0.5
- Input size: 640 × 640

5. Inference Pipeline

Load video file -> Read frames sequentially -> Perform bird detection using YOLO. -> Assign unique IDs using IOU-based tracking. -> Count unique tracked birds per timestamp. -> Estimate weight using bounding box area. -> Annotate frames with

bounding boxes, IDs, and counts. -> Save annotated video. -> Return JSON response via API.

6. Visual Results



7. Observations & Analysis

- Detection accuracy is high under stable lighting conditions.
- Tracking prevents double-counting of birds.
- Weight estimation is relative, not absolute.
- Occlusions and overlapping birds can reduce accuracy.
- Processing speed depends on video resolution and hardware.

8. Future Work

- Integrate depth sensors or calibrated cameras for true weight estimation.
- Improve tracking using DeepSORT or ByteTrack.
- Add real-time live stream processing.
- Train a poultry-specific detection model.
- Deploy system on edge devices for farm usage.

9. Object Detection in Streamlit http://127.0.0.1:8000/docs


```
{
  "frames_processed": 5421,
  "max_bird_count": 17,
  "avg_bird_count": 13.8,
  "avg_weight_proxy": 32450.6
}
```