Computer Vision Project: Bullseye Detection Using Deep Learning

🔷 Project Overview

This project focuses on detecting red bullseyes using computer vision techniques powered by deep learning. The final model performs well in real-time video streams and static test cases.

🔷 Dataset Preparation

Initial Approach

- Sourced an initial dataset of **800 images** from **Roboflow**.
- Applied **augmentation techniques** (rotation, brightness shift, etc.), increasing dataset size to **~2000 images**.
- However, this caused **overfitting**, reducing generalization.

Revised Strategy

- Reused the original **800 Roboflow images**.
- Added **50 diverse bullseye images** from **Kaggle**.
- Trained the model, but encountered **false positives**:
    - Circular objects (including faces) were often misclassified as bullseyes.

Solution to False Positives

- Created a "**negative class**" dataset with **85 non-bullseye images**.
- Retrained the model to distinguish bullseyes from similar-looking but irrelevant objects.

Camera Inconsistencies

- Model failed on live camera feed due to **noise and lighting variation**.
- Augmented the dataset by:
    - Varying exposure.
    - Adding artificial noise to existing images.
- This significantly improved real-time performance.

🔷 Training and Learning Experience

How I Trained the Model

- Followed **YouTube tutorials** that explained model training and commands.
- Used **YOLOv8** for object detection.
- Exported the trained model using: `model.export(format='onnx')`

Lessons Learned

- Initially tried to **retrain an already trained model (best.pt)** with "bad" images.
- This increased the loss and worsened performance.
- Realized that **it's better to retrain from scratch with a balanced dataset**.

🔷 Results

- Final model performs well in:
  - Static test images.
  - Live camera input.
  - Video feeds.
- Model successfully exported as ONNX for deployment:
  - `best.pt`

Git hub link - https://github.com/infinity2147/Assignments_2025

🎓 Tools & Frameworks Used

- **Ultralytics YOLOv8**
- **Python**
- **OpenCV**
- **Roboflow**
- **Kaggle**
- YouTube video link - https://github.com/infinity2147/Assignments_2025

Future improvements-

- Will add cropped images to the dateset .
- Will try to implement zoom in object detection (tiling + detection)

Author: *Anant Asati*