

Computer Vision Project: Bullseye Detection Using Computer Vision and Deep Learning

Introduction

Accurate detection of bullseye patterns is essential for tasks like autonomous navigation, target tracking, and precision analysis. This project focuses on developing a robust, real-time bullseye detection system capable of identifying red and white bullseye patterns across varying environmental conditions using computer vision and deep learning techniques.

Dataset Creation

❖ Data Collection

- Collected over 800 images of red and white bullseyes from:
 - Online sources
 - Public-dataset on Roboflow

❖ Diversity Considerations

- Lighting conditions: Bright light, low light,
- Angles: Frontal, tilted, partial visibility
- Backgrounds: Clean, cluttered, textured, outdoor
- Noise
- Horizontal and Vertical flips

Exported to yolov8 format and made annotations of bounding boxes for images using Roboflow only.

Model Training Workflow

- YOLOv8n (nano) model selected for faster inference & light-weight usage.
- data.yaml file made for yolo
- Used Ultralytics YOLOv8 library
- Trained for 3 epochs.

Model Evaluation and Results

Metric	Value
Precision (P)	0.999
Recall (R)	1.0
mAP@0.5	0.995
mAP@0.5:0.95	0.87

Interpretation:

- Extremely high Precision & Recall indicates almost no false positives or false negatives.
- mAP@0.5:0.95 of 0.87 indicates strong performance even at stricter IoU thresholds.
- However, with so much accuracy I thought data must be overfitting, So, I checked using some images not having bullseye and model didn't detect bullseye
- But, it detected bullseye of other colour like blue and white bullseye as my dataset didn't contain any image to train model that not to detect other colour.
- So, I made other model named 'best1.pt' using more better dataset such that it didn't detect any other colour bullseye.

Model Optimization for Real-Time

Conversion to ONNX using `model.export(format='onnx')`

(However, I am unable to convert to onnx format in my laptop due to some error of missing libraries)

Real-time Inference Without OpenCV: `model.predict(source=0, show=True, conf=0.5)`

PRESENTATION:

https://www.canva.com/design/DAGmkh9MPFI/oibJZMCcn4VWjmQubXRJYA/edit?utm_content=DAGmkh9MPFI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Work Report

April 5

Knowing about creating dataset, labelling and its other alternatives, watching their tutorials

April 6

Creating dataset, annotation

April 7

Find better dataset on roboflow, training and evaluation

April 8

Updating dataset using more images, re-training and evaluation, real time detection using yolo

Preyash

24B2184

