RAKSHAK
IIT BOMBAY

# Bullseye Detection

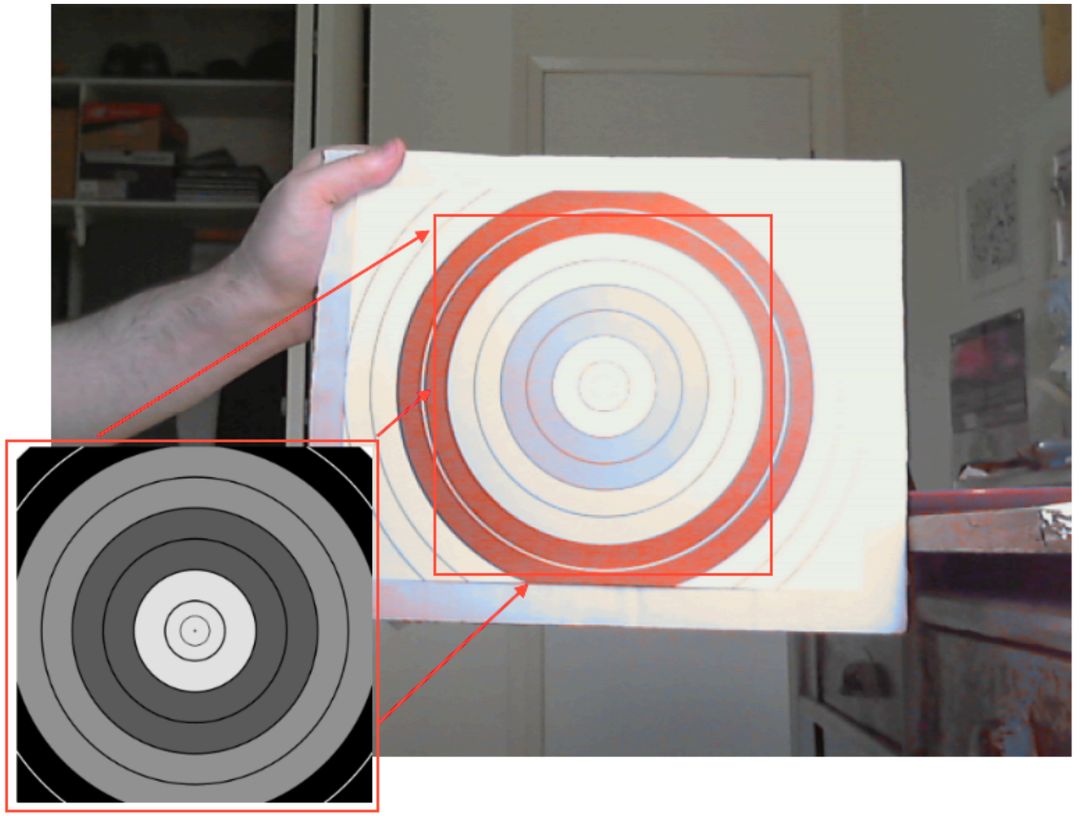
## Using Computer Vision and Deep Learning

PRESENTED BY: PREYASH SHAH

05-05-2025

# Problem Statement

- **Importance of Real-Time Bullseye Detection:** Accurate and efficient detection of bullseyes is crucial in various applications, such as robotics, gaming, and sports.
- **Limitations of Existing Generic Object Detectors:** Generic object detectors may not perform well on specific objects like bullseyes, which require precise detection.
- **Project Goal:** Develop a robust and accurate bullseye detection system using deep learning techniques.



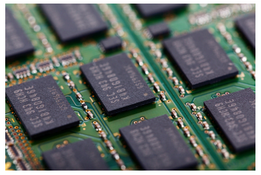
# Project Workflow

**Data Collection** Gathered diverse bullseye images

Labeled using LabelImg and Roboflow **Annotation**

**Model Training** Trained with Yolov8

Measured accuracy(mAP), precision **Evaluation**



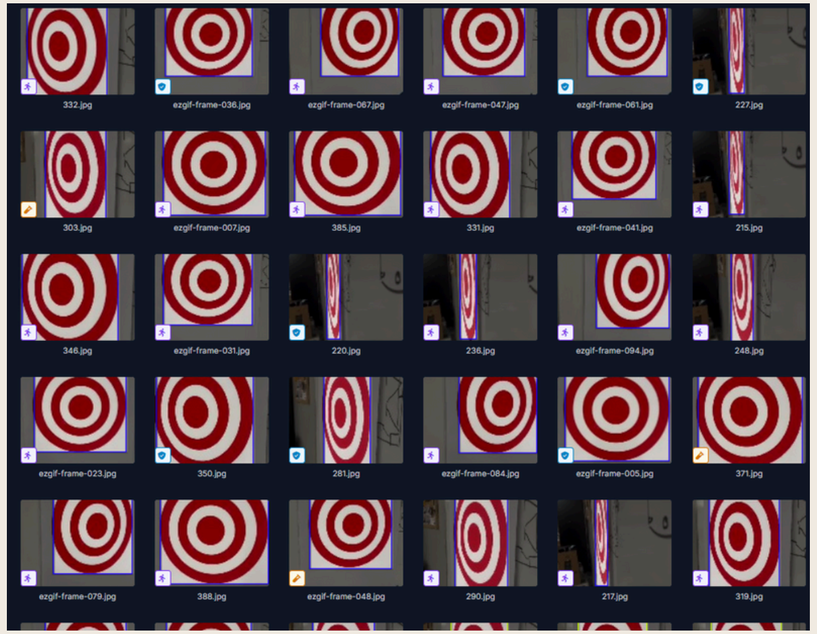
**Optimization** Converted to ONNX format

# Custom Dataset Creation

### ❖ Data Collection

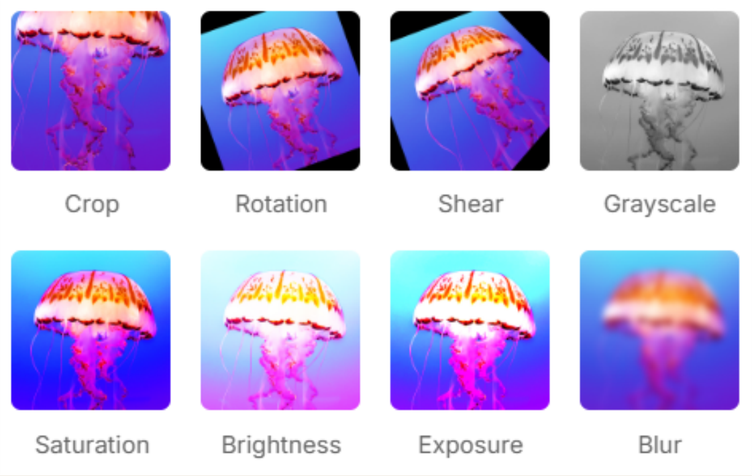
Collected over 800 images of red and white bullseyes from:

- 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




### ❖ Annotation Tool Used:

LabelImg tool was used to annotate the images.

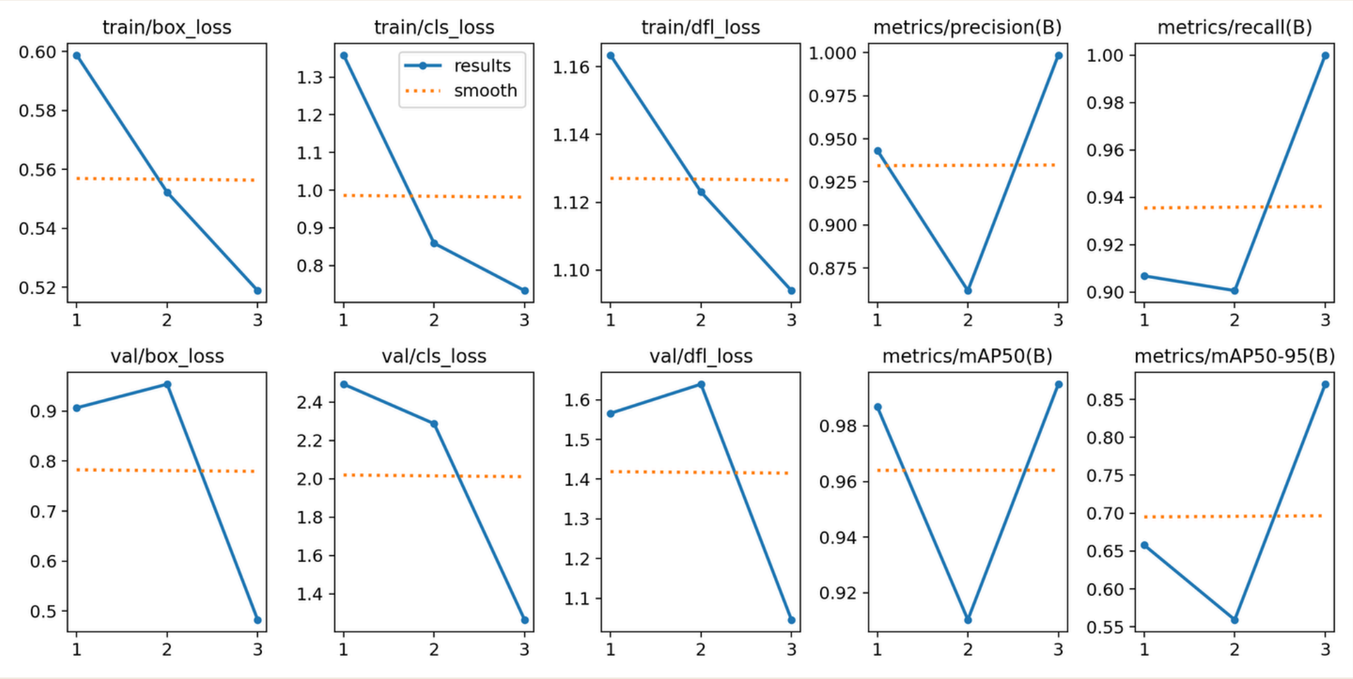
Exported to yolov8 format and made annotations of bounding boxes for images using Roboflow only.

# Model Training

YOLOv8n (nano) model selected for faster inference & light-weight usage.
data.yaml file made for yolo

- Training Details:
  - Epochs: 3 epochs were used for training.
  - Learning Rate:0.01 was used for training.
  - Batch Size: 16 was used for training.

Conversion to ONNX format




# Model Evaluation

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

**Command Snippet:**

```
model = YOLO("yolov8n.pt")
train_results=model.train(data="data.yaml", epochs=3)
model = YOLO('best.pt')
metrics = model.val()
```

# Challenges faced

- Difficulty in Sourcing Diverse Bullseye Images: Locating a varied selection of bullseye images proved to be quite challenging.
- Initially, the accuracy of the model appeared excessively high, raising concerns about potential overfitting.
- The model was also able to detect bullseyes in colors beyond those present in the dataset, which consisted solely of red and white images. Therefore, it is necessary to update the dataset to ensure it does not recognize bullseyes of different colors.
- I encountered issues installing the required packages, preventing conversion to the ONNX format; however, I possess the code necessary for this conversion.

## Key Findings:

- Creating custom Dataset
- **Real-time Inference without OpenCV**, using Yolo's support: model.predict(source=0, show=True, conf=0.5)
- Data augmentation, optimization with ONNX, YOLOv8, LabelImg, Roboflow

# Thank You