# Hackfest Machine Learning

This project aimed to detect **FireExtinguisher**, **ToolBox**, and **OxygenTank** using **YOLOv8**. We enhanced performance using **voting**, training multiple models and combining their outputs. The approach led to improved **mAP scores** and better overall robustness.

---

## Step-by-Step Methodology

### Dataset Preparation

- The dataset consisted of labeled images for three objects: FireExtinguisher, ToolBox, and OxygenTank. The dataset was split into train, validation, and test sets.
- To improve generalization and handle edge cases, data augmentation techniques such as random mosaic augmentation, horizontal flips, and brightness/contrast adjustments were applied.

### Model Training

We used the YOLOv8 framework due to its state-of-the-art performance and speed in object detection tasks. The project also experimented with YOLOv5 for model diversity in bagging. We explored nano, small,medium and large models.

Key hyperparameters used: Epochs, Batch Size, Learning Rate (lr0), Final LR (lrf), Momentum, Optimizer, Mosaic Augmentation Probability
Training was conducted on Kaggle's GPU environment with CUDA support enabled for accelerated training using PyTorch and the Ultralytics YOLO package.

### Evaluation Metrics

Model performance was primarily assessed using **mAP@0.5**
**Confusion Matrix** analysis showed reduced false positives and better class distinction.

## Visual Documentation

!python train.py --model yolov8m.pt --epochs 50 --batch_size 24 --optimizer Adam

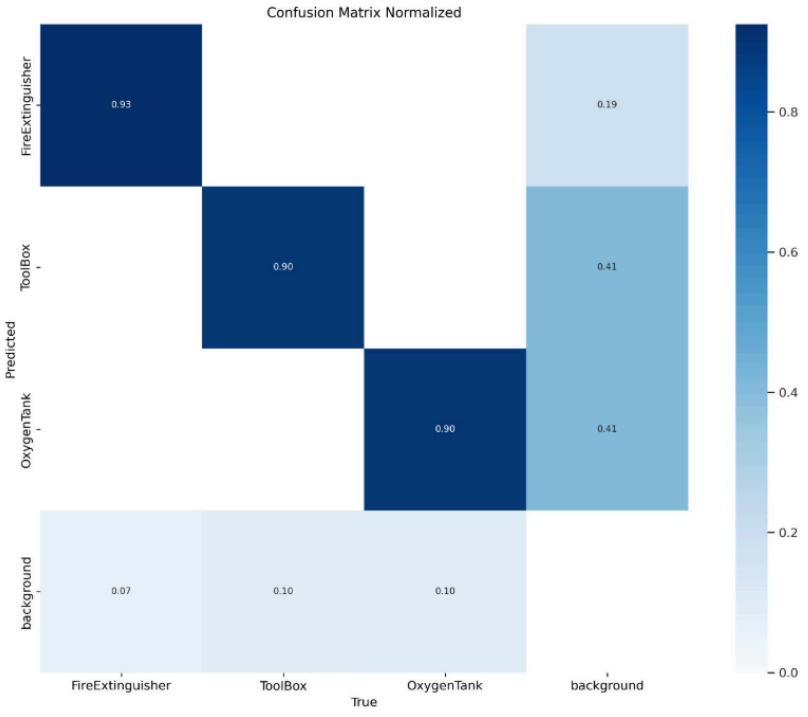
```
val: Scanning /kaggle/input/data/test/labels... 394 images, 0 backgrounds, 0 cor
val: WARNING ⚠ Cache directory /kaggle/input/data/test is not writeable, cache not saved.
                 Class     Images  Instances      Box(P          R      mAP50  m
                   all        394        549      0.928      0.769      0.858      0.757
      FireExtinguisher        179        179      0.938       0.81      0.876      0.739
               ToolBox        191        191      0.946      0.738      0.848      0.792
            OxygenTank        179        179      0.901       0.76      0.851       0.74
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered
in less
  xa[xa < 0] = -1
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered
in less
  xa[xa < 0] = -1
Speed: 0.8ms preprocess, 6.4ms inference, 0.0ms loss, 1.1ms postprocess per image
Results saved to runs/detect/val
```

!python train.py --model yolov8m.pt --epochs 5 --batch_size 24 --optimizer AdamW --lr0 0.001 --lrf 0.001 --momentum 0.9  --mosaic 0.4

| Class | Images | Instances | Box(P | R | mAP50 | m |
|---|---|---|---|---|---|---|
| all | 394 | 549 | 0.914 | 0.759 | 0.849 | 0.713 |
| FireExtinguisher | 179 | 179 | 0.933 | 0.804 | 0.869 | 0.704 |
| ToolBox | 191 | 191 | 0.918 | 0.757 | 0.848 | 0.756 |
| OxygenTank | 179 | 179 | 0.891 | 0.715 | 0.831 | 0.681 |

!python train.py --model yolov8l.pt --epochs 5 --batch_size 24 --optimizer AdamW --lr0 0.001 --lrf 0.001 --momentum 0.9  --mosaic 0.4

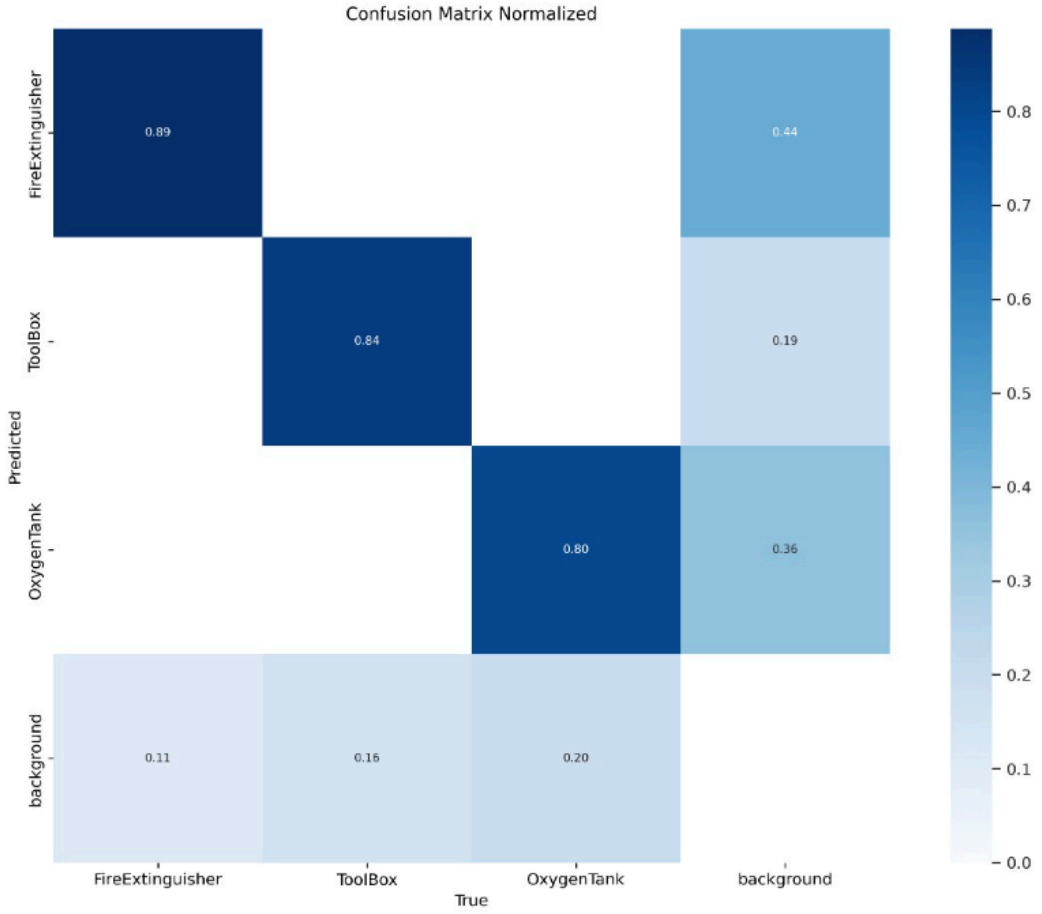
| Class | Images | Instances | Box(P | R | mAP50 | m |
|---|---|---|---|---|---|---|
| all | 394 | 549 | 0.827 | 0.734 | 0.8 | 0.669 |
| FireExtinguisher | 179 | 179 | 0.868 | 0.777 | 0.839 | 0.68 |
| ToolBox | 191 | 191 | 0.769 | 0.717 | 0.781 | 0.699 |
| OxygenTank | 179 | 179 | 0.844 | 0.709 | 0.78 | 0.628 |




!python train.py --model yolov5l.pt --epochs 5 --batch_size 24 --optimizer AdamW --lr0 0.001 --lrf 0.001 --momentum 0.9  --mosaic 0.4

| Class | Images | Instances | Box(P | R | mAP50 | m |
|---|---|---|---|---|---|---|
| all | 394 | 549 | 0.882 | 0.708 | 0.809 | 0.644 |
| FireExtinguisher | 179 | 179 | 0.938 | 0.761 | 0.836 | 0.653 |
| ToolBox | 191 | 191 | 0.84 | 0.733 | 0.813 | 0.693 |
| OxygenTank | 179 | 179 | 0.869 | 0.629 | 0.778 | 0.588 |

Best: !python train.py --model yolov8m.pt --epochs 30 --batch_size 24 --optimizer AdamW --lr0 0.001 --lrf 0.001 --momentum 0.9 --mosaic 0.4




| Class | Images | Instances | Box(P | R | mAP50 | m |
|---|---|---|---|---|---|---|
| all | 394 | 549 | 0.896 | 0.823 | 0.876 | 0.783 |
| FireExtinguisher | 179 | 179 | 0.884 | 0.883 | 0.891 | 0.772 |
| ToolBox | 191 | 191 | 0.933 | 0.8 | 0.877 | 0.803 |
| OxygenTank | 179 | 179 | 0.872 | 0.788 | 0.86 | 0.773 |

**Parameters:**
Model: `yolov8m.pt`, Epochs: `30`, Batch Size: `24`, Optimizer: `AdamW`, Initial Learning, Rate (lr0): `0.001`, Final Learning Rate Factor (lrf): `0.001`, Momentum: `0.9`, Mosaic Augmentation: `0.4`

`mAP50 score: 0.876`

# Challenges Faced

- Despite its popularity, the SDG variant yielded significantly lower mAP scores on our dataset, possibly due to overfitting or poor generalization.
- Numerous runtime and dependency errors prevented stable execution of the optimization loop, limiting fine-tuning capabilities.
- Bagging models trained on bootstrapped datasets did not yield performance gains, likely due to insufficient model diversity or overlapping errors.
- Efforts to resize, normalize, or batch process the large image set for preprocessing ran into performance bottlenecks and memory constraints, making it impractical for this experiment.

# Solutions Attempted & Lessons Learned

- After poor results with SDG, switching to AdamW provided better performance.
- With Optuna failing, hyperparameters were tuned manually through iterative experimentation, which, while slower, still allowed gradual improvements.
- Since bagging did not enhance results, we used voting, which turned out to be more efficient.