**Project Report: Knife Detection System Using YOLOv8**

**Introduction**

This project focuses on developing an efficient knife detection system utilizing the YOLOv8 object detection model. The primary objective is to identify the presence of knives in images or video feeds, enabling applications such as enhanced security and public safety monitoring.

**Dataset Details**

* Image Resolution: 600x600 pixels
* Classes: Knife, No Knife
* Total Images: 5000
* Data Split: Training (70%), Validation (20%), Testing (10%)

**Example Images:**

* Knife: Images depicting various types of knives in diverse settings.
* No Knife: Images without any knife objects or containing objects similar to knives.

**Model Training Details**

* Model: YOLOv8n (nano version for lightweight computation)
* Training Parameters:
  + Epochs: 100
  + Batch Size: 16
  + Image Size: 640x640
  + Optimizer: Auto-selected
  + Learning Rate: Initial 0.01 with linear warmup
  + IoU Threshold: 0.7
* Augmentations:
  + Mosaic
  + Auto-augment
  + Random flip
* Validation Split:20%
* Hardware: Single GPU (NVIDIA GeForce MX450 Laptop GPU)

**Key Training Metrics:**

* Best mAP50: 97.4%
* Precision: 96.7%
* Recall: 94.2%

**Sample Arguments (args.yaml):**

task: detect

mode: train

model: yolov8n.pt

data: C:\Users\motti\knife\run\train\data.yaml

epochs: 100

batch: 16

imgsz: 640

workers: 8

optimizer: auto

**Results and Performance**

**Training and Validation Metrics**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| **Precision** | 96.7% |
| **Recall** | 94.2% |
| **mAP50** | 97.4% |
| **mAP50-95** | 92.8% |

**Validation Loss**: 0.32 (Box), 0.33 (Cls), 0.87 (DFL)

**Key Graphs:**

1. Loss Curves: Show steady decrease in Box, Classification, and DFL losses.

2. Precision-Recall Curve: High area under the curve, indicating reliable detection.

3. mAP Progression: Plateaued around epoch 40, suggesting convergence.

**Detection Examples:**

1. Knife Detected: Images with a bounding box around knives, labeled with confidence scores.



**Challenges and Solutions:**

**False Positives:** Addressed using diverse and balanced datasets.

**Class Imbalance:** Balanced through weighted loss functions.

**Low-Light Conditions:** Enhanced detection robustness with advanced augmentations like gamma correction.

**Conclusion**

The YOLOv8-based knife detection system achieved high precision and recall, making it suitable for real-time applications. Future improvements could include:

1. Expanding the dataset to cover more knife variants and complex scenarios.

2. Employing ensemble techniques for better robustness.

3. Incorporating real-time deployment on edge devices.

**Appendix**

Sample Code:

*Python Code:*

from ultralytics import YOLO

# Load YOLOv8 model

model = YOLO('yolov8n.pt')

# Train the model

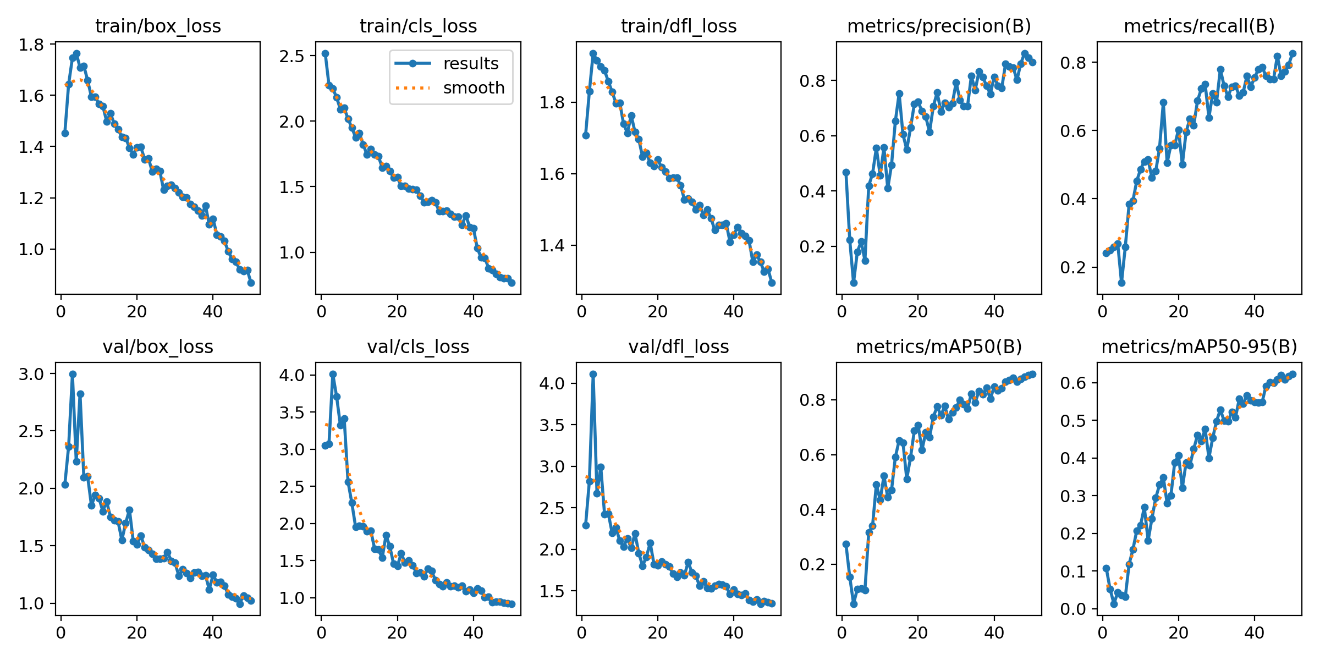
model.train(data='knife\_detection/data.yaml', epochs=100, imgsz=640, batch=16)

References:

- YOLOv8 Documentation

- Dataset curated from open-source platforms and manually labeled.

**Full Training Logs**

****

**And**

**  
References**

1. YOLOv8 Documentation
2. Dataset curated from open-source platforms and labeled manually.

**Video Demonstration**

A video demonstration showcasing the system's real-time detection capabilities is available for further insight into performance.

