Knife Detection Using YOLOv8s and YOLOv11s

name	id
Omar Mohammed Omar Abu Al Rub	160208
Asem Omar Hmoud Al Hattamleh	163633
Abdulraheem Musa Abdulraheem Alnatoor	160324

Course: Computer Vision – Fall 2025/2026
Course instructor: Dr. Abdullah Al-amaren

Knife Detection Using YOLOv8s and YOLOv11s

Abstract

Our project develops and evaluates a real time knife detection system using single stage object detection . We experimented with multiple modern YOLO architectures , including YOLOv8n , YOLOv8s , YOLOv8m , and YOLOv11n , YOLOv11s , YOLOv11m which were fine tuned on a knife detection dataset with a single class that is knife . The training pipeline was implemented using the Ultralytics framework with transfer learning from pretrained weights , standard augmentations , and mixed precision training . Performance was assessed using standard object detection metrics like Precision , Recall , mAP@0.5 , and mAP@0.95. Both models achieved strong accuracy on the held out evaluation set , with YOLOv8s and YOLOv11s delivering comparable performance while differing in computational cost . Finally , the trained model was integrated into a Streamlit web interface to demonstrate practical deployment for interactive inference

1.Problem Statement

Knife detection is a **safety oriented** computer vision task aimed at identifying the presence of knives in images or video frames . The objective of our project is to train a robust detector that can localize knives with bounding boxes and achieve high accuracy under real world variability (different lighting , backgrounds , scales , and occlusions) , while remaining efficient enough for near real time inference

---

2.Research on Neural Networks and Architectures

Object detection has progressed from two stage detectors region proposal and classification to efficient one stage detectors direct prediction of bounding boxes and class probabilities . YOLO family detectors are widely used due to their balance of speed and accuracy

In our work

- **YOLOv8n** uses an ultra lightweight architecture designed for very fast inference and minimal resource usage
- **YOLOv8s** uses a lightweight architecture optimized for fast inference while maintaining accuracy
- **YOLOv8m** uses a larger architecture designed to improve detection accuracy with higher computational cost
- **YOLOv11n** uses a compact architecture optimized for high efficiency and fast inference
- **YOLOv11s** is a newer YOLO variant designed to improve efficiency and feature representation
- **YOLOv11m** uses a larger architecture designed to enhance detection performance with increased computation

From our models summarie

- YOLOv8n: **~3.2M parameters** , **~8.7 GFLOPs** (imgsz=512)
- YOLOv8s: **~11.14M parameters** , **~28.6 GFLOPs** (imgsz=512)
- YOLOv8m: **~25.9M parameters** , **~79.3 GFLOPs** (imgsz=512)
- YOLOv11n: **~2.9M parameters** , **~7.5 GFLOPs** (imgsz=512)
- YOLOv11s: **~9.43M parameters** , **~21.5 GFLOPs** (imgsz=512)
- YOLOv11m: **~22.1M parameters** , **~60.8 GFLOPs** (imgsz=512)

3.Models Development and Training

3.1 Dataset Selection and Exploration

A single class dataset was used with the class list defined in our data.yaml as

- **Class names:** knife

From the training logs we can see that

- **Train set:** 7,351 images (7,483 labeled boxes)
- **Validation set:** 918 images (937 labeled boxes)

Notes in our data : The Ultralytics loader reported some mixed detection segmentation annotation structure and therefore we **used bounding boxes only** (segments were removed)

Table 1. Dataset summary

Split	Images	Instances (boxes)	Classes
Train	7,351	7,483	1
Validation	918	937	1
Test	920	940	1

3.2.Training and Validation Process

We trained both models for **60 epochs** with transfer learning from pretrained weights . Automatic Mixed Precision (AMP) was enabled

Table 2. Key training hyperparameters our best run

Item	YOLOv8s	YOLOv11s
Image size	512	512
Batch size	32	32
Epochs	60	60
Augmentation	Enabled	Enabled
Optimizer	Auto selected AdamW (lr = 0.002, momentum = 0.9)	Auto selected AdamW (lr = 0.002, momentum = 0.9)
Early stop	15	15
AMP	Enabled	Enabled

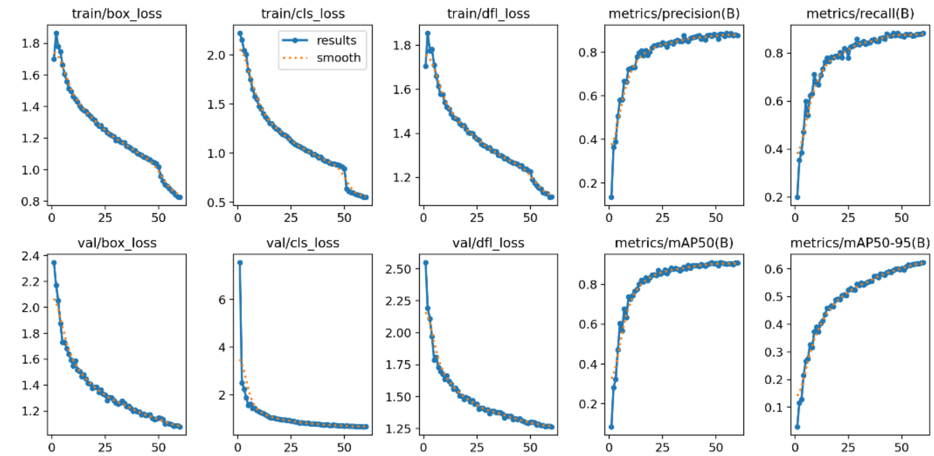


Figure 1. yolov8results.png

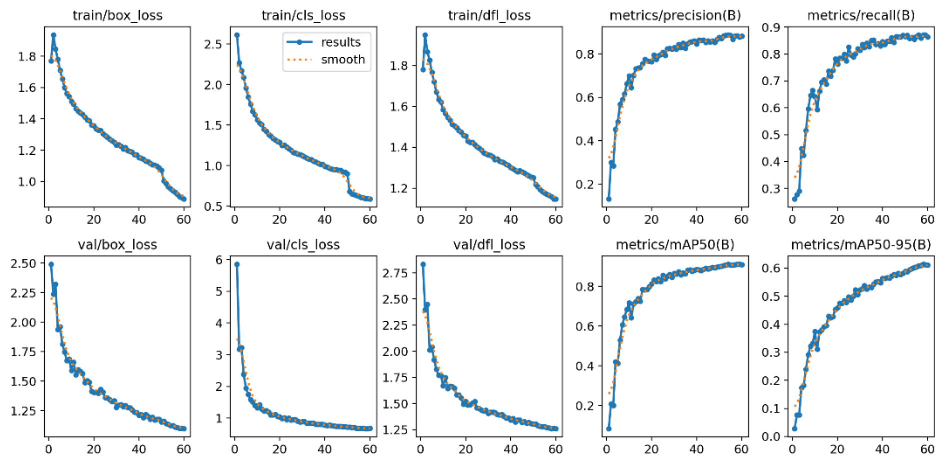


Figure 2. yolov11results.png

Table 3.Hyperparameter Search Results

Table 3 summarizes our validation performance for the hyperparameter configurations explored for each architecture . Each setting varies image resolution and batch size while keeping the learning rate request and augmentation enabled for consistency . our BEST row indicates the top performing configuration per model based on validation metrics

Table 3.Hyperparameter search results (validation)

Architecture	Setting ID	imgsz	batch	lr0	aug	Val P	Val R	Val mAP50	Val mAP50-95
YOLOv8s	S1	512	16	0.01	on	0.846	0.866	0.893	0.612
YOLOv8s	S2	640	64	0.01	on	0.855	0.865	0.901	0.624
YOLOv8s	**BEST**	**512**	**32**	**0.01**	**on**	**0.864**	**0.877**	**0.910**	**0.631**
YOLOv11s	S1	512	16	0.01	on	0.847	0.857	0.897	0.601
YOLOv11s	S2	640	64	0.01	on	0.857	0.867	0.907	0.613
YOLOv11s	**BEST**	**512**	**32**	**0.01**	**on**	**0.866**	**0.867**	**0.905**	**0.6163**

4.Testing and Evaluation

4.1.Evaluation Metrics

Performance is reported using

- **Precision (P)** and **Recall (R)**
- **mAP@0.5** (IoU threshold 0.5)
- **mAP@0.95** (average across IoU thresholds 0.5 to 0.95)

4.2.Test Results (Best Models)

Table 4.Best model performance on testing set

Model	Test P	Test R	Test mAP50	Test mAP50-95
YOLOv8s	0.8784	0.8684	0.8987	0.6014
YOLOv11s	0.8784	0.8630	0.8999	0.6100

4.3.Overfitting Discussion

From our learning curves (Figures 1–2), validation metrics rise steadily and stabilize toward the end of training , indicating the training procedure achieves convergence without obvious late stage collapse in validation performance. Additionally the project used :

- data augmentation
- pretrained initialization
- and early stopping patience (15)

which collectively reduce overfitting risk .

4.4.Results Analysis and Model Choice

Both detectors achieve very similar test performance

- YOLOv11s shows slightly higher **test mAP50-95** (0.6100 vs 0.6014)
- while YOLOv8s shows slightly higher **validation mAP50** in the best run (0.910 vs 0.905)

From an efficiency standpoint , YOLOv11s is lighter in compute (**~21.5 GFLOPs vs ~28.6 GFLOPs**) and has fewer parameters , which can be advantageous for deployment. Given the small accuracy differences , the final choice can reasonably be driven by deployment constraints (latency and compute budget)

Dataset Size Ablation YOLOv11s

To analyze how dataset size affects performance , a dataset-size sweep was run on the best model (YOLOv11s) , producing the following plots :

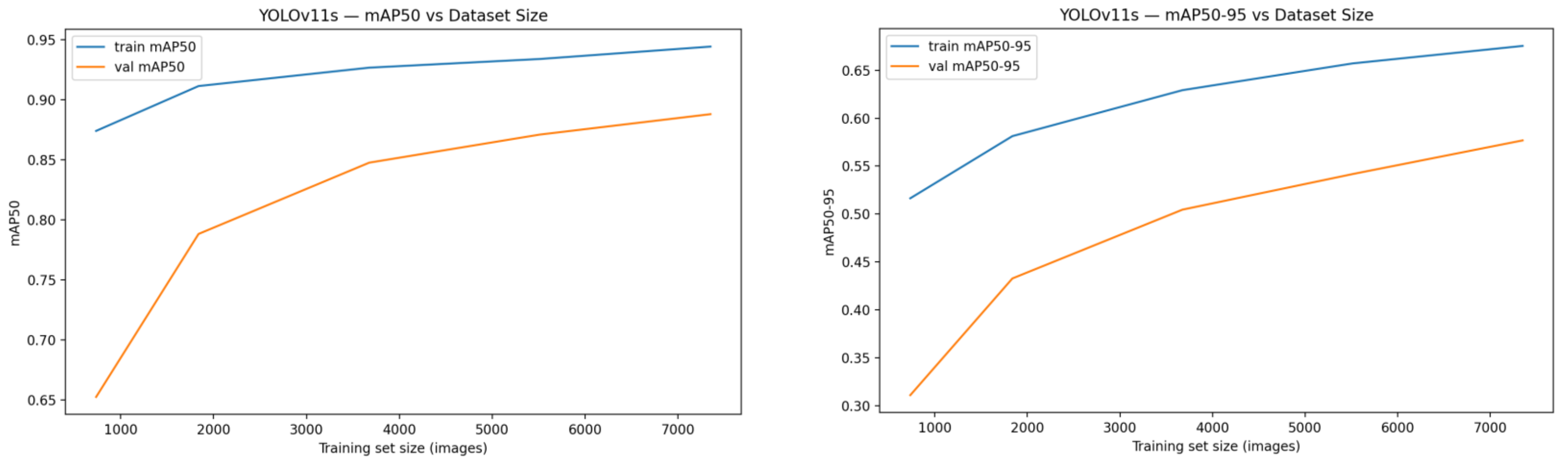


Figure 3&4 validation versus dataset

Table 5. Dataset size sweep summary YOLOv11s

Train Fraction	Train Images	Val mAP50	Val mAP50-95	Train Time (min)
0.10	735	0.750	0.452	7.58
0.25	1,837	0.805	0.503	12.98
0.50	3,675	0.845	0.545	22.17
0.75	5,513	0.872	0.564	32.71
1.00	7,351	0.888	0.577	42.83

Computational Requirements

All experiments were executed in the Kaggle environment with **the NVIDIA Tesla T4 GPUs (T4×2)** available. Training runs were configured with standard Ultralytics settings and AMP enabled. For the best runs

- **YOLOv8s training time :** ~82.5 minutes (60 epochs , imgsz=512 , bs=32)
- **YOLOv11s training time :** ~85.8 minutes (60 epochs , imgsz=512 , bs=32)

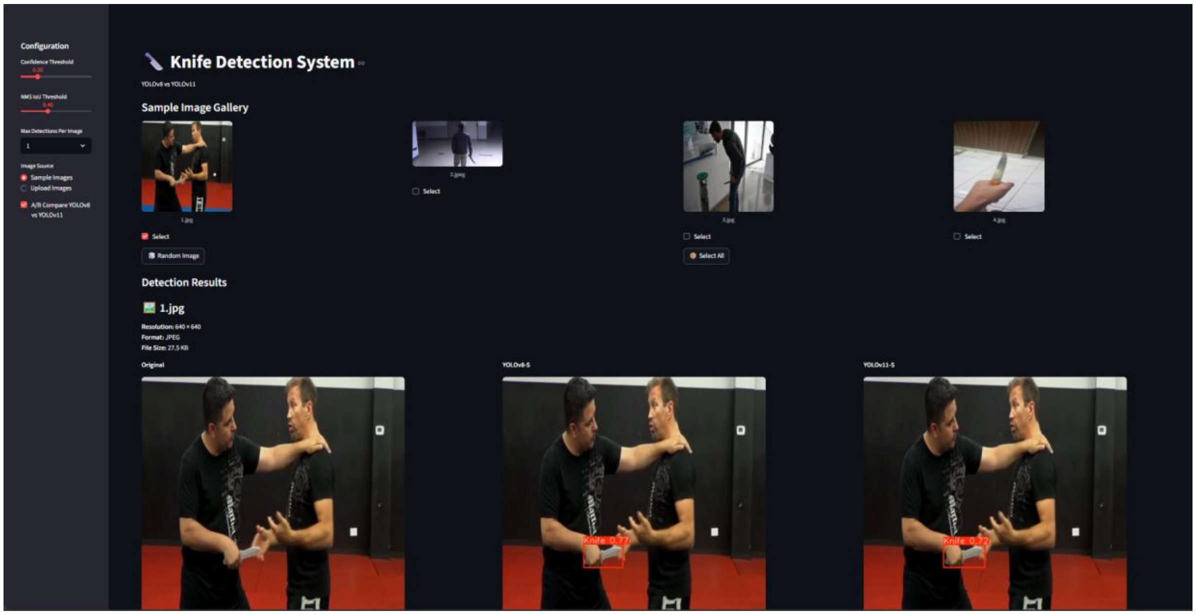
Reported per-image inference speed from validation output:

- YOLOv8s: ~8.8 ms inference / image (imgsz=512)
- YOLOv11s: ~20.0 ms inference / image (imgsz=512)

4.5 Interface Development

A Streamlit interface was developed to demonstrate interactive knife detection.

Our deployed interface allows users to upload media and run inference using the trained YOLO model . The application overlays bounding boxes and confidence scores on detected knives and provides a practical demonstration of the end to end workflow from model training to real world usage .



4.6 Critical Evaluation

Key limitations and risks

1. **Dataset bias :** performance may degrade in unseen environments (different camera angles , lighting , motion blur , crowded scenes)
2. **Single class scope :** the model only detects knives and may confuse visually similar objects (tools , shiny utensils) depending on training coverage
3. **Annotation format warning :** mixed detect/segment artifacts were observed so the training proceeded using boxes only , which is appropriate for pure detection but should be cleaned in future iterations
4. **Operational thresholding :** real world deployments typically require calibration of confidence thresholds to control false positives

Future improvements:

- add negative or hard negative examples
- expand to multi-class on the weapon categories
- evaluate on video sequences
- and perform robustness testing (night , low res CCTV , occlusions)

References

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016. *You only look once: Unified, real-time object detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M., 2020. *YOLOv4: Optimal speed and accuracy of object detection*. arXiv preprint arXiv:2004.10934

Ultralytics, 2025a. *Ultralytics YOLOv8 Models Documentation*. [online] Available at: https://docs.ultralytics.com/models/yolov8/ [Accessed 27 December 2025]

Ultralytics, 2025b. *Ultralytics YOLO11 Models Documentation*. [online] Available at: https://docs.ultralytics.com/models/yolo11/ [Accessed 27 December 2025]

Ultralytics, 2025c. *Train mode — Model training with Ultralytics YOLO*. [online] Available at: https://docs.ultralytics.com/modes/train/ [Accessed 27 December 2025]

Ultralytics, 2025d. *Object detection task documentation*. [online] Available at: https://docs.ultralytics.com/tasks/detect/ [Accessed 27 December 2025]

Ahmed, M. & Khan, S., 2021. *Knife detection in images using YOLOv3*. In: Proceedings of the 2021 International Conference on Image Processing and Vision (ICIPV). IEEE, pp. 1–6

Streamlit Inc., 2023. *Streamlit documentation*. [online] Available at: https://docs.streamlit.io/ [Accessed 27 December 2025]

Custom Knife Detection Dataset, 2025. *Knife detection dataset with 7,351 training images, 918 validation images and 920 test images*