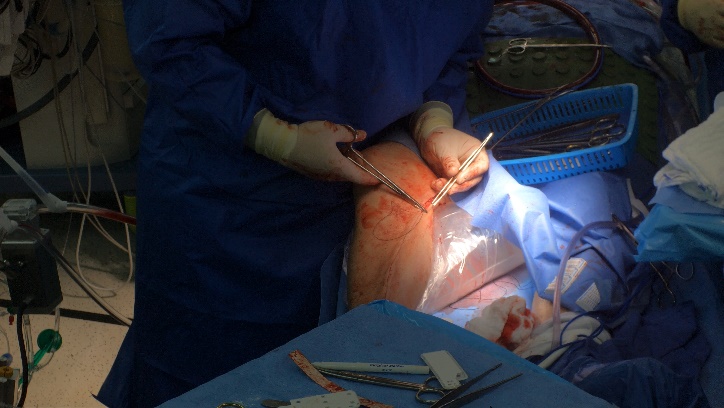
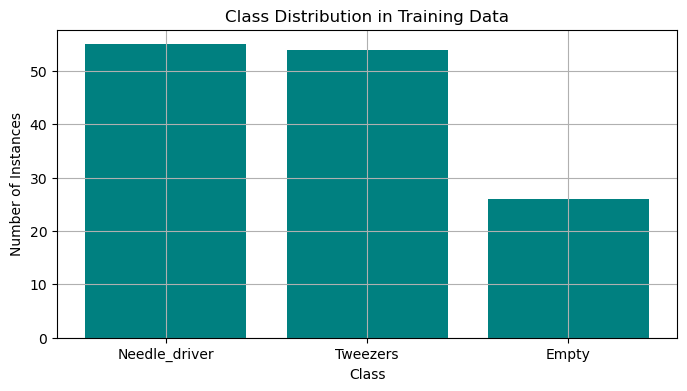
**Computer Vision, Surgical Applications 0970222 -HW1 Report**

1. **Exploratory Data Analysis**
   1. **Image Visualization (with/without label)**  
       ****To understand the data visually, We also labeled images from the training set and overlaid the bounding box to understand the labels better:A person in surgical gown performing surgery

      AI-generated content may be incorrect.A person with surgical scissors on their arm

      AI-generated content may be incorrect.
   2. **Insights from looking at the data:**After viewing several labeled images from the training set, we made the following observations:
      * Most images are close-up views of leg suturing.
      * Handsand tools are sometimes occluded or cut off, making detection harder.
      * Lighting varies — some images are bright, while others have shadows or glare.
      * Tools are often overlapping or in realistic positions, as in actual surgeries.

These observations suggest that, although the dataset is small, it captures realistic surgical variability—including occlusions, lighting differences, and visual clutter—which makes detection more challenging but also can helps the model generalize better to unseen surgical scenarios, such as the OOD (out-of-distribution) video.

* 1. **Data Distribution analysis  
     Training Data:** **Total labeled images:** 61, **Total bounding boxes:** 135

**Average boxes per image:** 2.21  
**Most frequent class:** Needle\_driver, **Least frequent class:** Empty  
**Validation Data:**  
A graph with a number of blue squares

AI-generated content may be incorrect.  
**Total labeled images:** 10, **Total bounding boxes:** 22

**Average boxes per image:** 2.20

**Most frequent class:** Needle\_driver, **Least frequent class**: Empty

1. **Experiments**
   1. **BaseLine Model  
      Data Loading, Pre-processing, and Cleaning:**  
      We used the provided labeled dataset, Data was loaded using the Ultralytics YOLO framework, which supports direct ingestion of YOLO-formatted annotations.  
      **Training Techniques:**We trained a YOLOv8n model as our baseline. YOLOv8n was chosen for its lightweight architecture and fast training times, which enabled quicker iterations. The initial training configuration was:
      * **Epochs: 50**
      * **Image size: 650 pixels**
      * **Batch size: 16**
      * **Learning rate: 1e-3**

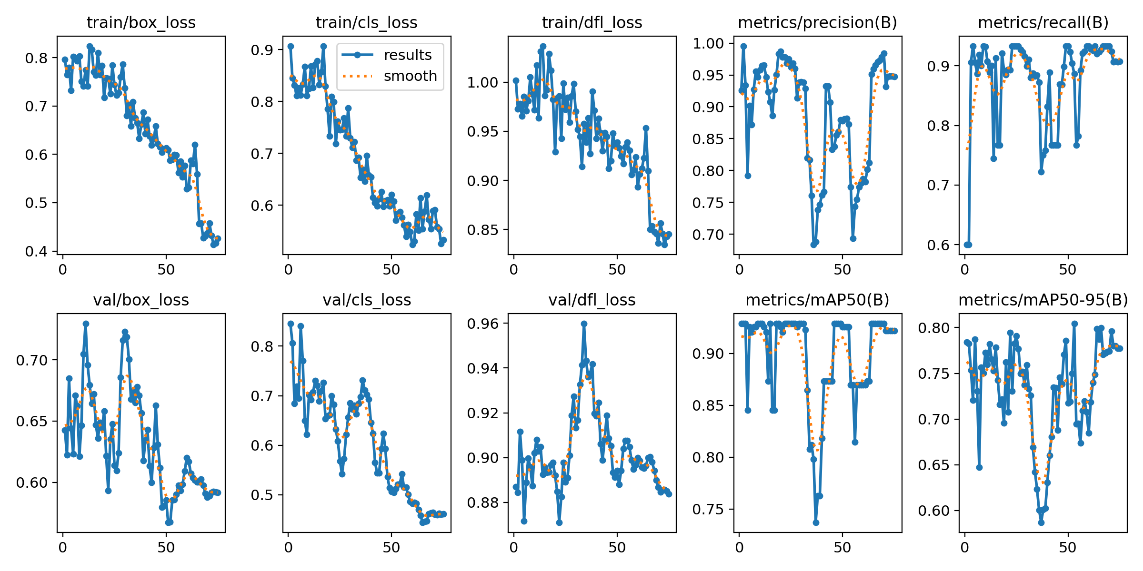
This baseline served as a reference point for further optimization and pseudo-labeling.

**Regularization:**To address potential overfitting, especially given the small dataset size, we incorporated L2 regularization via the weight\_decay parameter. Various weight\_decay values were tested during hyperparameter tuning to assess their impact on generalization.

**Hyperparameter Tuning:**We used Optuna, a framework for automated hyperparameter optimization. The tuning process involved running multiple short training trials (10 epochs each) to evaluate different configurations. The following hyperparameter ranges were explored:

After selecting the most promising combinations, we trained the model for 75 epochs using the best configuration found:

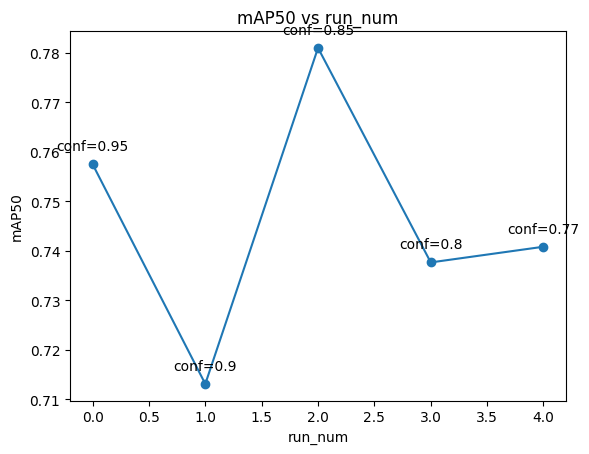
* **Learning rate (lr0):** 0.00135
* **Momentum:** 0.925
* **Weight decay:** 0.00013
* **Batch size:** 8

This tuning significantly improved performance, as evaluated using mean Average Precision (mAP) on the validation set.  
****

* 1. **Psuedo label model IID**  
     For the pseudo-labeling stage, we used the in-distribution (ID) videos. To reduce redundancy and avoid overfitting due to highly similar frames, we sampled every 10th frame rather than using all frames.

We used the trained baseline model to generate pseudo-labels for the ID video frames. To refine the quality of these pseudo-labels, we experimented with several confidence thresholds to filter out uncertain predictions. The thresholds tested were:  
[0.95, 0.9, 0.85, 0.8, 0.77]

These pseudo-labeled frames were then used to further train the model in a semi-supervised fashion.

For confidience level 0.85 we insepct the highest mAP50 therefore we chose this model for the IID videos.

* 1. **Psuedo label model OOD**Data preprocessing – nothing special

1. **Discussion and Conclusions**
   1. **Baseline Model:**
      * Demonstrated that even a minimal dataset (61 labeled images) can produce a functional object detector when paired with an efficient architecture (YOLOv8n).
      * While initial mAP scores were promising, the model exhibited limitations in challenging visual conditions (e.g., occlusions, variable lighting), revealing a lack of robustness.
      * Ultimately served as a strong foundation but lacked the generalization capacity needed for more diverse surgical environments.
   2. **IID Pseudo-Label Model:**
      * Semi-supervised learning via pseudo-labeling substantially expanded the effective training set without any additional manual annotations.
      * Selecting every 10th frame from in-distribution videos provided visual diversity while avoiding redundancy, allowing the model to learn from real surgical dynamics.
      * Confidence threshold tuning was critical: a threshold of 0.85 balanced label precision with quantity, leading to the best detection performance (mAP50).
      * The enhanced training set improved model generalization and resilience to visual noise, validating pseudo-labeling as a viable strategy for low-resource domains.
      * However, the process introduced some risk of label noise and error propagation, highlighting the need for careful threshold calibration and visual inspection.
   3. **OOD Model:**