Sina Project : Surgical Instrument Detection

Project Phases & Tasks

Phase 1: Literature and Dataset Survey

Tasks:

- **Understand the Core Concepts:**

Write a clear and concise explanation of the difference between Image Classification, Object Detection, and Instance Segmentation.
For each concept, provide a hypothetical example related to cataract surgery (e.g., "This image contains a phacoemulsifier probe" vs. "The phacoemulsifier probe is at these coordinates" vs. "These exact pixels belong to the phacoemulsifier probe").

- **Literature Search**

Using academic search engines (Google Scholar, Papers with Code, arXiv), find 1-2 key research papers on surgical instrument detection. You don't need to read them in-depth, but understand the common challenges (e.g., motion blur, occlusion, reflection) and the datasets they use. (here you can see the datasets and state of the art models, focus on those tables)

- **Dataset Search and Analysis:**

Your primary task is to find a suitable public dataset for cataract surgery instrument detection.
Where to look: Google Scholar, Papers with Code, Kaggle, Roboflow Universe, arXiv.
Search keywords: "cataract surgery instrument dataset," "ophthalmic tool detection," "surgical instrument tracking dataset."
Deliverable: A short report comparing at least two datasets you found. For each dataset, analyze:
Name & Source: (e.g., CATARACTS Challenge 2017, Cholec80, a dataset from Roboflow Universe).
Size: Number of images/videos, number of annotated frames.
Annotation Type: Does it have bounding boxes (perfect for detection) or segmentation masks?
Classes: What instruments are labeled? (e.g., Forceps, Phacoemulsifier, Cannula).
License: Is it free for academic/personal use?
Accessibility: Is it easy to download?

- **Final Selection and Justification:**

Based on your analysis, choose one dataset for the project.
Justify your choice. (e.g., "I chose Dataset X because it has high-quality bounding box annotations for 10 instrument classes and is well-documented, making it ideal for a YOLO-based detection task.")

Phase 2: Establish a Baseline with YOLO

- **Train the Baseline Model:**

Start with a small, fast model to get a quick result. We recommend yolov8n.pt (the "nano" version).
Train the model for a reasonable number of epochs (e.g., 50 epochs).
Use a standard image size (e.g., 640x640).
Crucially, monitor and record your resources during training.

- **Evaluate and Report:**

The YOLO training script will output performance metrics. Record the key metrics for your baseline.
Deliverable: A summary of your baseline performance. This should be a clear, concise report containing:
Model Used: YOLOv8n
Metrics:
    mAP50 (mean Average Precision at IoU=0.50)
    mAP50-95 (mean Average Precision over IoU thresholds from 0.50 to 0.95)
Resource Usage:
    Total Training Time: (e.g., 1 hour 15 minutes)
    Peak VRAM Usage: (Use nvidia-smi to monitor this during training)
    Model Size: (The final .pt file size in MB)
    Number of Parameters: (e.g., 3.2 million for YOLOv8n)
Qualitative Results: Include 3-5 example images from your validation set with the model's predicted bounding boxes drawn on them.

Phase 3: Hypothesis-Driven Improvement

Task:

Choose at least two hypotheses from the categories below (or create your own). For each hypothesis:

1. **State the Hypothesis:** Clearly articulate what you expect to happen (e.g., "Hypothesis: Increasing the input resolution will improve the detection of small instruments, leading to a higher mAP, but at the cost of increased VRAM and training time.").
2. **Design the Experiment:** Describe exactly what you will change (e.g., "I will re-train the model with the imgsz parameter set to 1280 instead of 640.").
3. **Run the Experiment & Collect Data:** Re-train the model and collect the same metrics as in Phase 2.
4. **Analyze and Conclude:** Compare the results to your baseline in a table. Was your hypothesis supported or rejected? Why do you think you got these results?

Examples of Diverse Hypotheses for Improvement:
Hypothesis (Model Size): "A larger, more complex model like YOLOv8s or YOLOv8m has more capacity to learn complex features. I hypothesize it will achieve a significantly higher mAP than the YOLOv8n baseline, but with a linear increase in VRAM and training time."
Hypothesis (Data Augmentation): "Surgical videos often have glare and lighting changes. I hypothesize that increasing the intensity of color and brightness augmentations (e.g., hsv_s, hsv_v parameters in YOLO) will make the model more robust to these variations and improve mAP without significantly affecting training time."
Hypothesis (Input Resolution): "Some instruments, like needle tips, are very small. I hypothesize that training at a higher resolution (e.g., imgsz=1024) will allow the model to 'see' these small objects better, improving precision and recall for smaller classes, at a significant VRAM cost."
Hypothesis (Training Duration): "My baseline model's loss curve was still decreasing after 50 epochs. I hypothesize that training for more epochs (e.g., 100 epochs) will allow the model to converge better and will increase mAP, with a risk of overfitting."
Hypothesis (Efficiency - Creative Bonus): "For real-time deployment, inference speed is critical. I hypothesize that I can export my trained model to a TensorRT format and achieve a 2x-3x speedup in inference with only a minimal (1-2%) drop in mAP compared to the PyTorch baseline."

Final Deliverable:

A final presentation or report summarizing the entire project.
A brief recap of Phase 1 and 2.
A detailed section for Phase 3, presenting each hypothesis, experiment, and conclusion.
A final comparison table showing the results of your baseline and all your experiments side-by-side (Model, mAP50, Training Time, VRAM, Inference Speed, etc.).