Master of Technology in AI Systems Project

PATTERN RECOGNITION SYSTEM PRACTICE MODULE

Project: Computer Vision System for Identifying Anatomical Structures in Surgical Operation Videos

REPORT

GROUP NO.: 9

GROUP MEMBERS:
ARSHI SAXENA - A0331999J

NORBERT OLIVER - A0328685M

PRANJALI SONAWANE - A0326167B

SHARVESH SUBHASH - A0327428Y

1. Executive Abstract

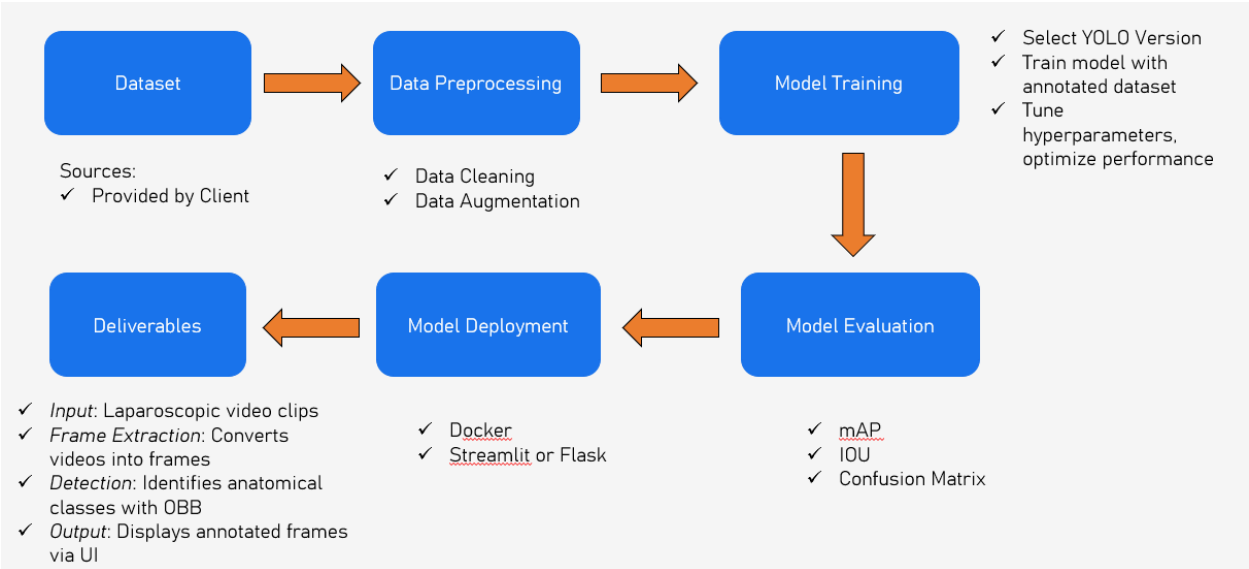
This project develops and rigorously evaluates an AI system that automatically identifies six critical anatomical structures in laparoscopic Transabdominal Preperitoneal (TEP) hernia surgery: triangle of pain, testicular vessels, vas deferens, inferior epigastric vessels, triangle of doom, and pubic bone. Using a curated, surgeon-annotated dataset of 5,200 images provided by NUS Medicine, we will fine-tune a modern YOLOv11-based detector (with oriented bounding boxes) to improve upon a prior YOLOv8 baseline in both accuracy and inference speed. The system targets two immediate use cases: (1) surgical education overlaying clear, consistent labels on operation videos to help trainees recognise anatomy reliably; and (2) a foundation for future intraoperative guidance where fast, accurate recognition is essential for avoiding high-risk zones.

The work of this project will primarily focus on dataset audit and preparation (quality checks, removal of unused labels, class-balance strategies, and targeted augmentation), followed by model training and hyperparameter tuning. Performance will be measured with mAP (per class and overall), IoU, confusion matrices, and latency. The best model will be containerised and deployed with a lightweight UI (Flask/Streamlit) that ingests video, extracts frames, runs detection, and visualises results with OBB overlays prioritising clarity, reproducibility, and ease of use for stakeholders. This project also serves as a foundation that can be further extended in terms of scope with exploration of Vision-Language Models and other techniques.

Expected outcomes are: (i) a documented accuracy and speed gain over YOLOv8, (ii) an end-to-end, containerized application suitable for classroom demos and pilot evaluations. Governance considerations such as data privacy, ethical use, and safe UI messaging will be observed, ensuring the solution remains aligned with clinical relevance, safety standards, and practical applicability.

2. System Design

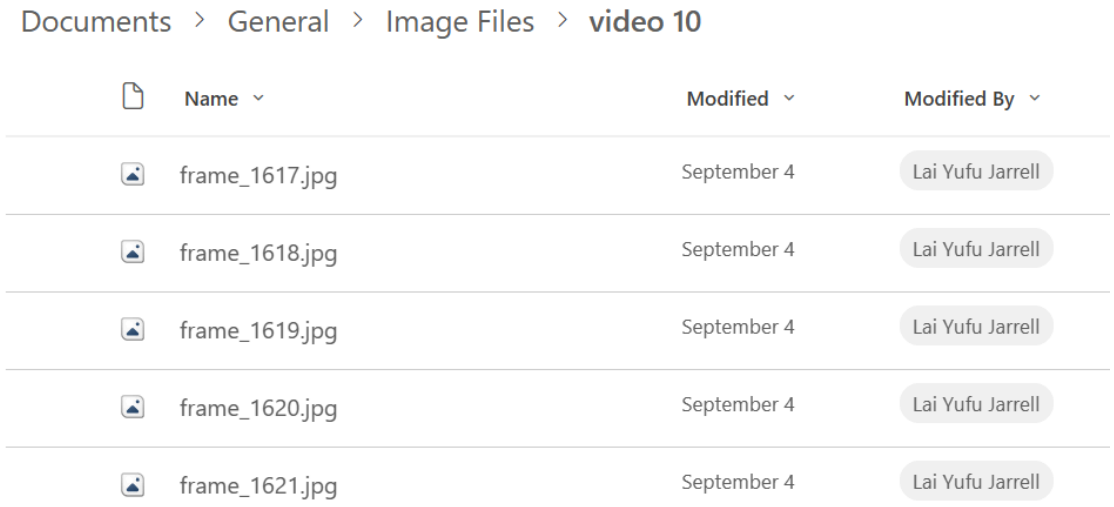
The complete system design based implementation template for different experiments that were carried out is the exact same mentioned in this diagram.

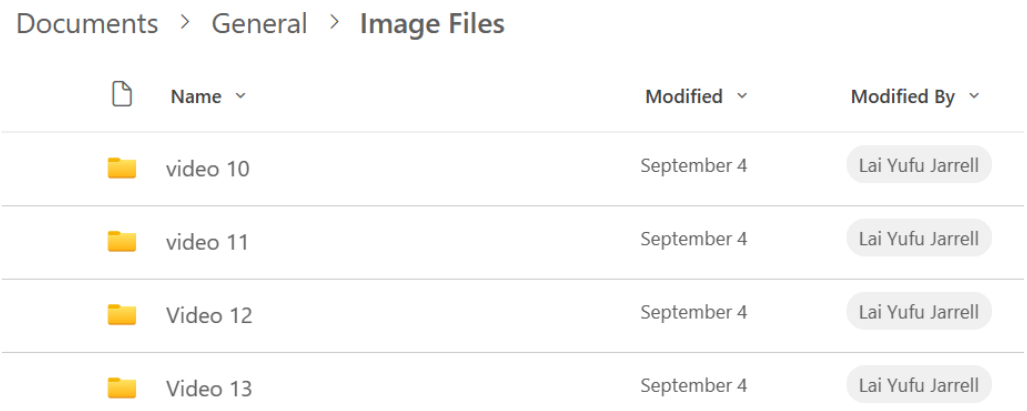


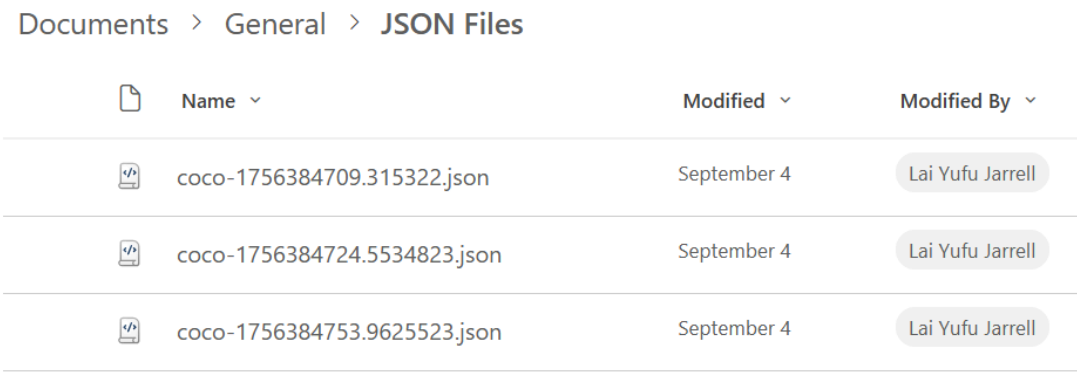
But the preliminary experiments had different configurations in their data-preprocessing techniques, to evaluate the immediate impact of preprocessing in performance of the models. All of the other experiments done for model comparison had a standardized data pre-processing pipeline that is fixed with different models for training and then evaluation. This has ensured that all of our experiments are reproducible and research publication worthy.

2.1 Client raw dataset cleaning and analysis

Our client from NUS Medicine had shared the raw annotated dataset with COCO segmentation format







As per requests from the client, we will be requiring classification for only 6 main classes, hence we performed cleaning.

The process of cleaning the dataset involves:

- Merging the coco annotations files into one coco annotation file.
- Remove annotation and category entries in the original dataset that is not part of the 6 wanted categories.
- Deduplicate image entries in the original dataset that point to the same image frame.
- Remove annotation entries in the original dataset that point to missing image frames.

Here are some observations that we have taken while in the process of cleaning:

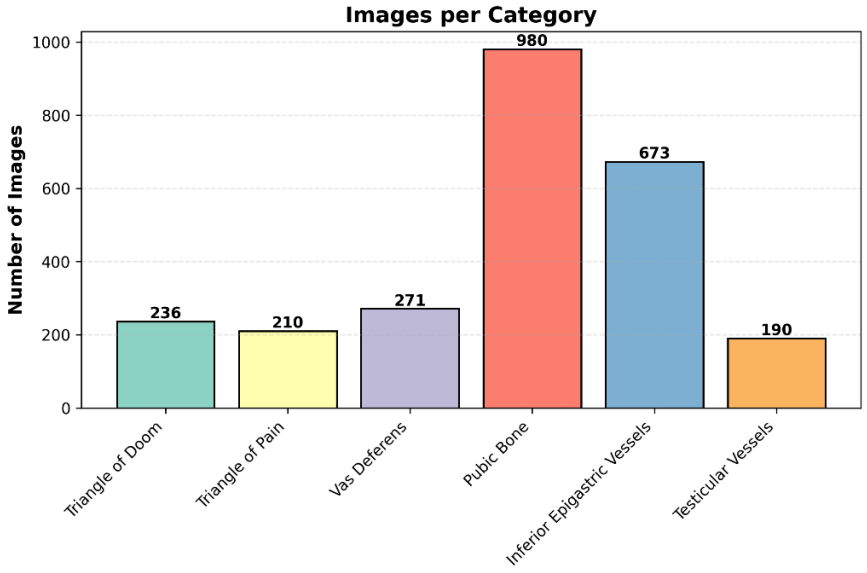
- There are only 1946 annotated image frames in the original dataset.
- 124 annotation entries in the original dataset were not part of the 6 wanted categories.
- 26 image entries in the original dataset refers to non-existent image frames.
- There are 376 duplicate annotation entries (have different boundary boxes, but refer to different image entries that point toward the same image frame).
- After cleaning, there are only 1420 out of 1946 image frames that we can use for our purposes.
- There are 98 annotation entries tied to 40 image frames, all of them having a resolution of 720x480, that contain out of bounds boundary boxes for their annotations.

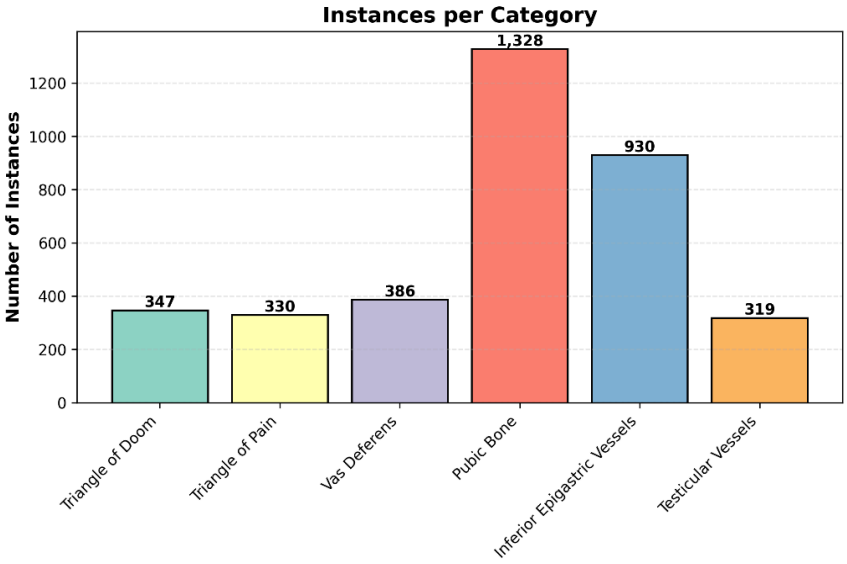
Here is the breakdown of statistics after analyzing the cleaned dataset.

- Category distribution:

Category	Associated Images	Instances
Triangle of Doom	236	347
Triangle of Pain	210	330
Vas Deferens	271	386

Pubic Bone	980	1,328
Inferior Epigastric Vessels	673	930
Testicular Vessels	190	319
Total	**1,420** (unique images)	**3,640**



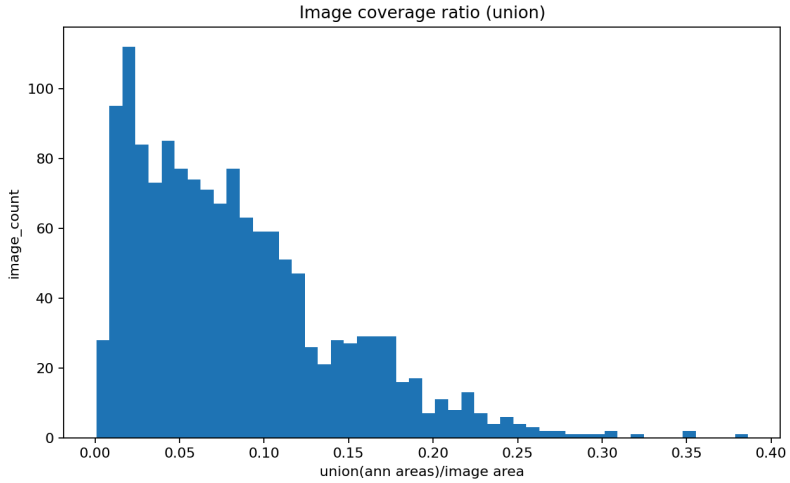


- Clear class imbalance: “Pubic Bone” dominates (980 images, 1,328 instances) and “Inferior Epigastric Vessels” is next (670,930). “Testicular Vessels” class is the scarcest (≈190/319).

- Instances per image hover ~1.3-1.7 across classes, suggesting mostly single (occasionally multiple) occurrences per frame.
- The ratio between the majority class against the minority class is 4.10, meaning that the data distribution is highly skewed towards one side (the majority).
- Interpretation: frequency skew will bias models toward Pubic Bone / IEV; rarer classes (e.g., Testicular Vessels) will be harder.

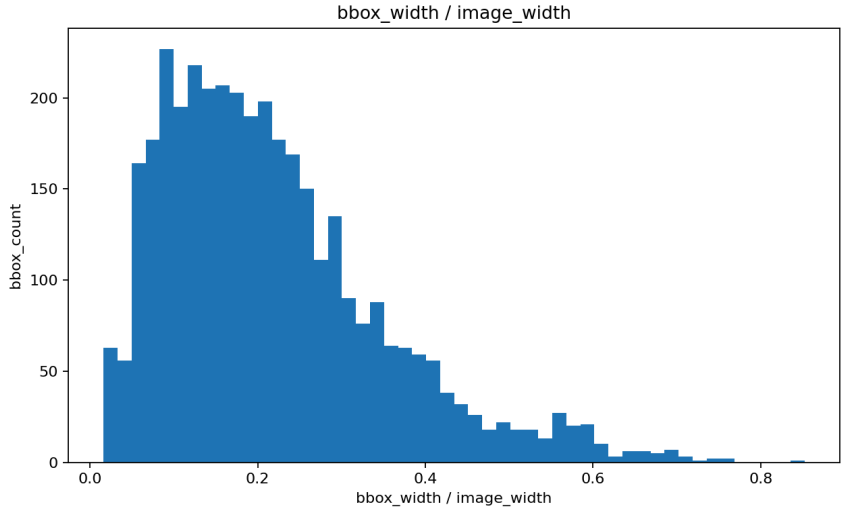
We resolved this Class imbalance before performing the main set of experiments.

- Coverage ratio (Annotation Area / Image area) distribution :

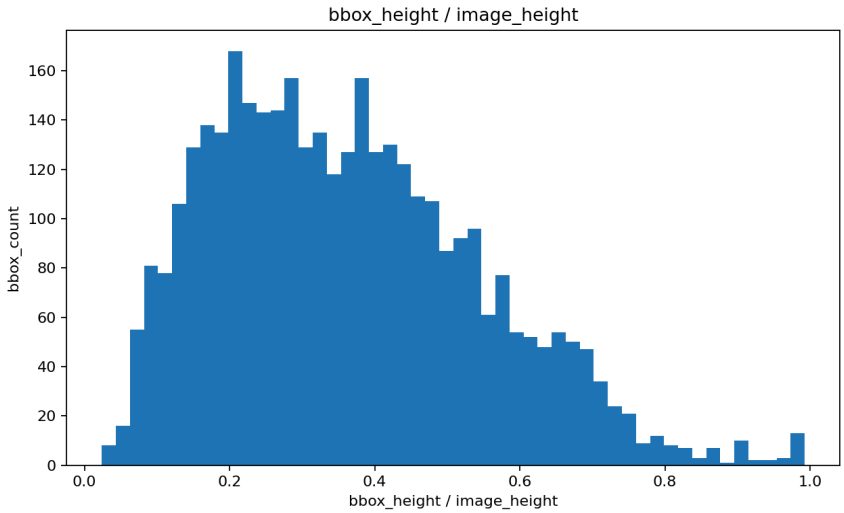


Most images have low coverage (<0.15), with a long but thin tail up to ~0.4.
Interpretation: per-image annotated area is sparse; backgrounds dominate most frames.

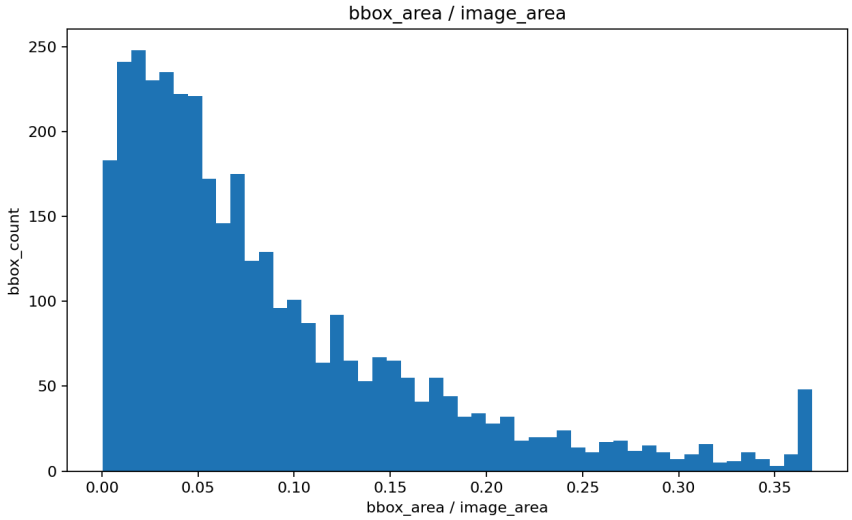
- Bounding Box based distribution



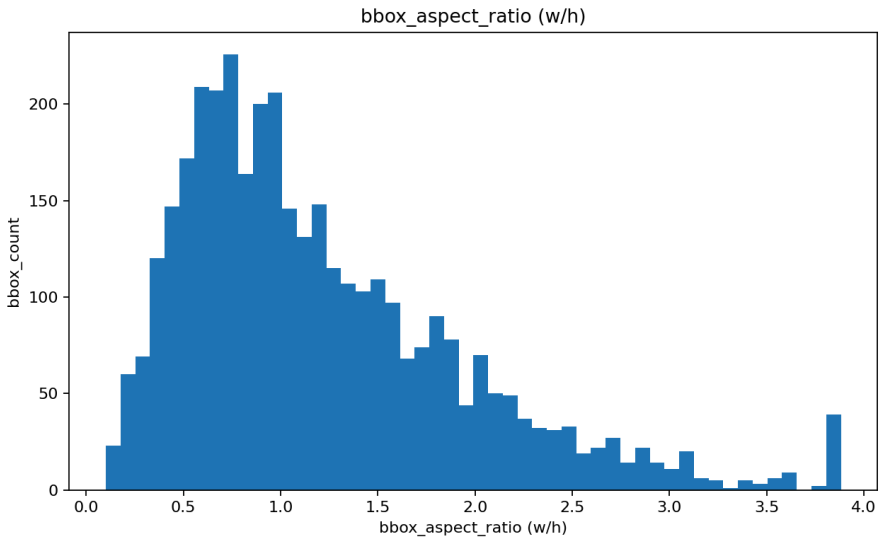
Concentration around ~0.1 - 0.35, tapering past 0.5 and rarely near 1.0.
Interpretation: widths are usually smaller than half the frame; very wide objects are uncommon.



Concentration around ~0.2–0.5, tapering toward 1.0.
Interpretation: box heights are typically moderate (not full-height), with occasional large/tall objects.



- Strong left-skew: most boxes cover only ~1–10% of the image; a long tail reaches ~0.37.
Interpretation: objects are generally small relative to the frame; “small-object” behavior will dominate.



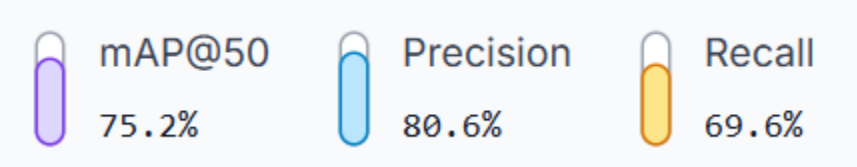
-> Bulk of boxes sit near 0.6 - 1.2 (roughly square to slightly wide), with a long right tail up to ~3.8 and a thinner left tail (<0.5).
Interpretation: shapes are varied, but mostly not extremely elongated, but some very wide boxes exist.

2.2 Preliminary Experiments

The preliminary set of experiments were done using the roboflow platform with a moderate performing version of Yolov11-seg mentioned in roboflow as “Accurate” category models. We performed two experiments here, one is without data augmentation and the other is with the minimum data augmentation.

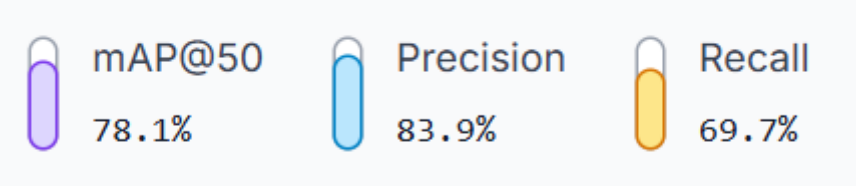
2.1.1 Without Data Augmentation

Here the same exact raw dataset after resizing the images into 640x640 dimension were given in as input to the MS COCO dataset pretrained Yolov11-seg model mentioned in the “Accurate” section of Roboflow.
Model training and evaluation were performed to record the performance of the model. Here are the results:



2.1.2 With Data Augmentation

Here, after the resizing of the images into 640x640 dimension, 3 image outputs were generated per training example of the dataset by randomly selecting any of the augmentation techniques mentioned below:
90° Rotate: Clockwise, Counter-Clockwise, Upside Down
Crop: 0% Minimum Zoom, 20% Maximum Zoom
Blur: Up to 2.5px
Noise: Up to 1.97% of pixels.
These images were given as inputs to the Yolov11-seg model mentioned in the "Accurate" section of Roboflow. Exact same model as the previous experiment for comparison. Model training and evaluation were performed to record the performance of the model. Here are the results:



Conclusion: Even with minimum data augmentation, we were able to improve the performance by 3%, with no changes in the hyperparameters of the Yolov11-seg model present in the "Accurate" section of the Roboflow platform which is described to be less accurate than the Large and medium sized versions of Yolov11-seg.

2.3 Data Preprocessing

2.3.1 Performing data augmentations

Through the preliminary experiments, we got the inference of improvement in performance with data augmentation techniques, therefore, we increased the data augmentation techniques. After the images were preprocessed with "Auto-Orient", they were resized to 640x640 dimensions. 5 output images were generated from each of the training example where each of 5 images are outputs of the randomly selected augmentation techniques listed below:
Flip: Horizontal, Vertical
90° Rotate: Clockwise, Counter-Clockwise, Upside Down
Crop: 0% Minimum Zoom, 20% Maximum Zoom
Rotation: Between -14° and +14°
Shear: ±15° Horizontal, ±15° Vertical
Grayscale: Apply to 21% of images
Hue: Between -25° and +25°
Saturation: Between -34% and +34%
Brightness: Between -25% and +25%
Exposure: Between -15% and +15%
Blur: Up to 2.5px
Noise: Up to 1.97% of pixels
After applying these techniques, we got 5396 images in our complete dataset, but these are still class imbalanced as observed from the previous class distribution graphs.

2.3.2 Handling class imbalance

Clearly from the raw data analysis section, we saw that the classes are imbalance, meaning the ratio between the highest represented class("MaxClassCount") vs the lowest represented class("MinClassCount") was 4.7, this is a terrible distortion for the training dataset, we need a comparable size of each class for good feature representation for the model to identify all of the different classes.

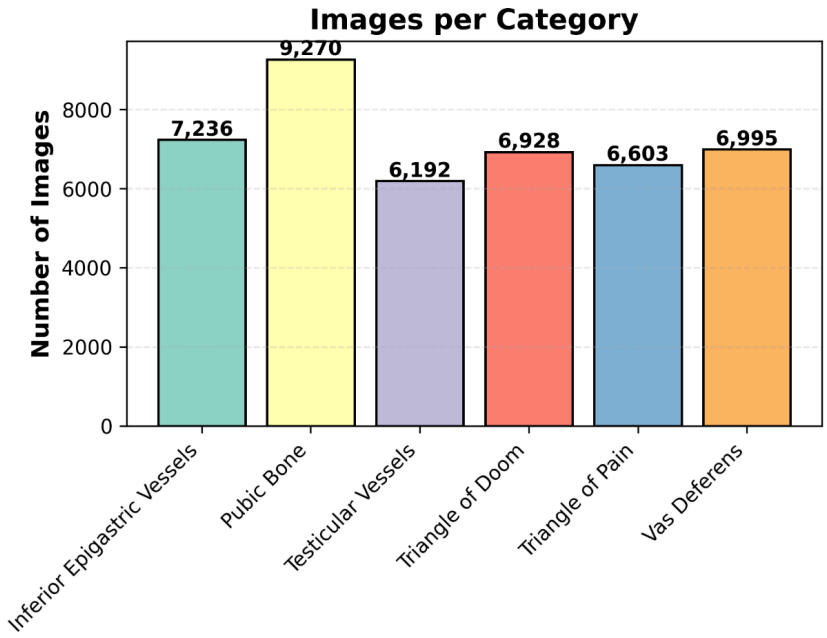
To resolve this issue, we used the oversampling technique to increase the number of instances of the least presented class by duplication of images.
By performing this iteratively for all of the classes with extra focus on the least represented classes, the number of images in the dataset increased from being 1420 to 15,387 images and the number of instances increased from 3640 to 67,372 instances. This increased the number of images and instances in each class and led to the ratio of "MaxClassCount" vs "MinClassCount" to 1.42, leading to a significantly less skewed dataset, meaning the class imbalance is resolved to the maximum possible extent.

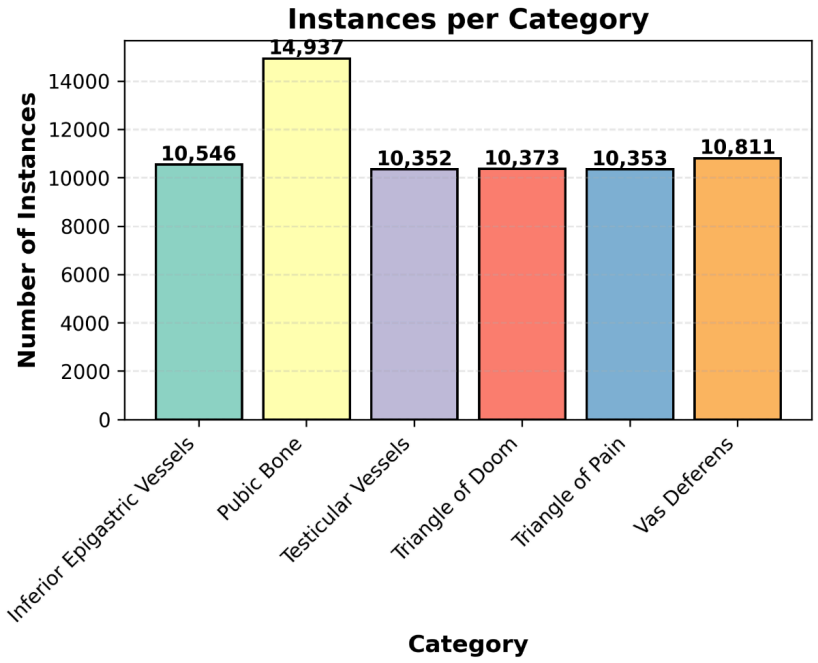
2.3.3 Analysis of the pre-processed dataset

Here is the breakdown of statistics after analyzing the balanced dataset that we will use to train our model.

Category	Associated Images	Instances
Inferior Epigastric Vessels	7,236	10,546
Pubic Bone	9,270	14,937
Testicular Vessels	6,192	10,352
Triangle of Doom	6,928	10,373
Triangle of Pain	6,603	10,353
Vas Deferens	6,995	10,811
Total	**15,387** (unique images)	**67,372**

- Class/Category Distribution:





- Through augmentation and duplication, we increase the dataset to have 10 times more images and 18 more instances than the original cleaned dataset.
- Previously, most classes significantly lack representations and are severely imbalanced. Now, most classes have a similar number of instances (around 10,300 to 10,800).
- Previously, the ratio between the majority class against the minority class was 4.10. Now the ratio has been reduced to 1.42, meaning that the data distribution is less skewed towards one side.
- Using this new dataset, the trained model would be less likely biased towards certain classes as the model would have a more generalized learning.

2.4 Model Training and Deployment

Here, for model training, we used two platforms. One is the google Collab platform and the other is the Roboflow platform. The Roboflow platform was used for performing memory intensive tasks like data augmentation, and conducting the model training for preliminary experiments. And the Google collab platform was used to train the main comparison study based experiments.
The model training function had the early stop feature to ensure the best performance, relative to the threshold loss metrics, is recorded and stored.

For model deployment, we use Docker to build an image which provides the necessary environment to run our application system. The application's user interface is in the form of a web application which internally uses the trained local model. By doing this, we can spin up as many containers as we want from the image, allowing us to deploy the model on servers or individual computers. A major benefit of this style of deployment is that they are very accessible and don't require an internet connection to run.

2.5 UI integration and Software output video:

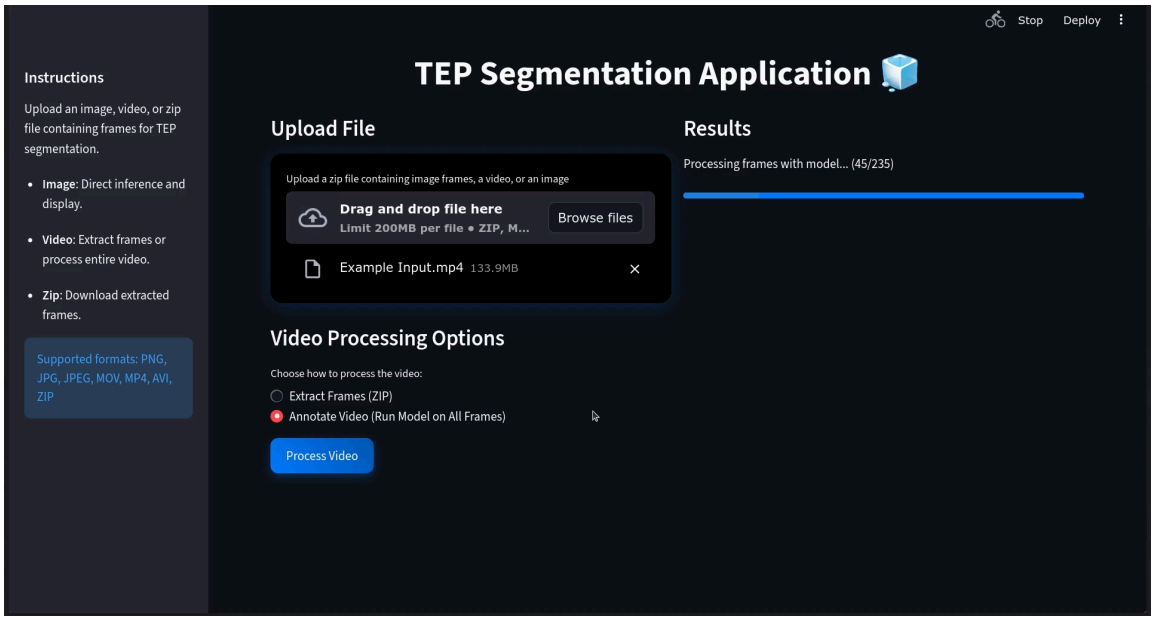
For the UI implementation, we use several core libraries:

- Streamlit to help create the web application interface.
- Moviepy to extract frames from input videos and combine output frames into videos.
- Pillow to add predicted label text on top of detected objects.

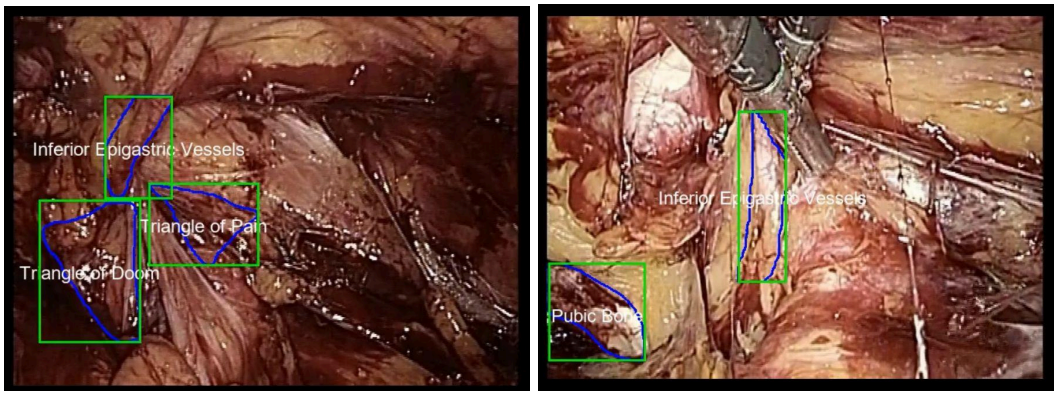
Features that the current UI support are:

- Upload an image to get the annotated version of the image.
- Upload a video to obtain all the video frames in the form of a zip file (the user can then selectively upload images to be annotated).
- Upload a video to obtain a new video that has been entirely annotated.

User interface for uploading and annotating a video:



Sample image frames from the output annotated video:



3. System Performance Validation and results

3.1 Briefing on Evaluation Metrics

Per-class, one-to-one matching: Predicted instances are sorted by confidence and matched to ground-truth (GT) instances of the same class using **IoU**. Each GT can match at most one prediction.

TP / FP / FN: A prediction is a **TP** if its IoU with an unmatched GT ≥ chosen threshold; otherwise it's a **FP**. Any GT left unmatched is a **FN**.

Precision: "Out of all the predicted positives, how many were actually right?"
Recall: "Out of all the true positives, how much did the model find ?"
F1: Harmonic mean of P and R.
AP → Average Precision

AP@t → Area under the PR curve at IoU threshold t (e.g., AP@0.50, AP@0.75)
AR → Average Recall
AR@I → Average recall over IoUs with at most 1, 10, or 100 detections per image.

For masks vs boxes:

- **Box metrics** use IoU of bounding boxes.
- **Mask metrics** use IoU of binary masks (after thresholding the predicted mask prob map).

For any predicted region P and ground truth G,

$$\mathrm{IoU}(P, G) = \frac{|P \cap G|}{|P \cup G|}$$

It's the fraction of *overlap* (intersection) out of the *total covered area* (union). Ranges 0–1.

Now we compare **two binary masks**:

- M_{gt}: pixels that belong to the object (ground truth)
- M_{pred}: pixels that the model says belong to the object

Each is like a black–white image: 1 = object pixel, 0 = background.

$$\mathrm{IoU}_{\mathrm{mask}} = \frac{|M_{\mathrm{gt}} \cap M_{\mathrm{pred}}|}{|M_{\mathrm{gt}} \cup M_{\mathrm{pred}}|}$$

- **Intersection** = pixels that **both** GT and prediction say are object.
- **Union** = pixels that **either** GT or prediction say are object.

- GT box: a rectangle (x_1, y_1, x_2, y_2)
- Predicted box: another rectangle

Then:

$$\mathrm{IoU}_{\mathrm{box}} = \frac{\text{area of intersection of the 2 boxes}}{\text{area of union of the 2 boxes}}$$

- **Intersection** = overlapped rectangular area.
- **Union** = area of box A + area of box B – intersection.

Diagrams of mathematical details are mentioned above for reference.

Overall, we will be using the following:

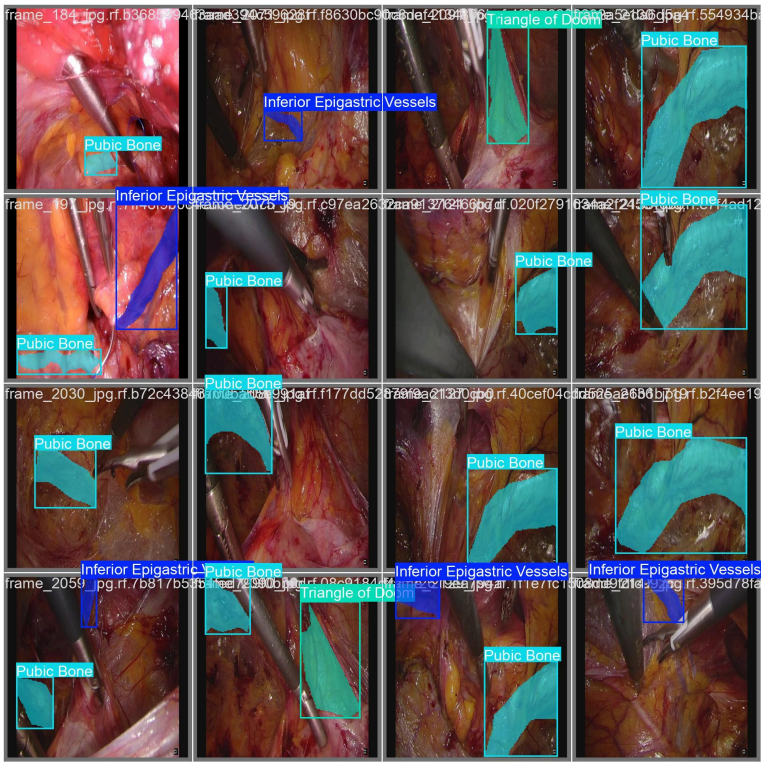
Box: Precision, Recall, **mAP50 (bbox)**, **mAP50-95 (bbox)**.

Mask: Precision, Recall, **mAP50 (mask)**, **mAP50-95 (mask)**.

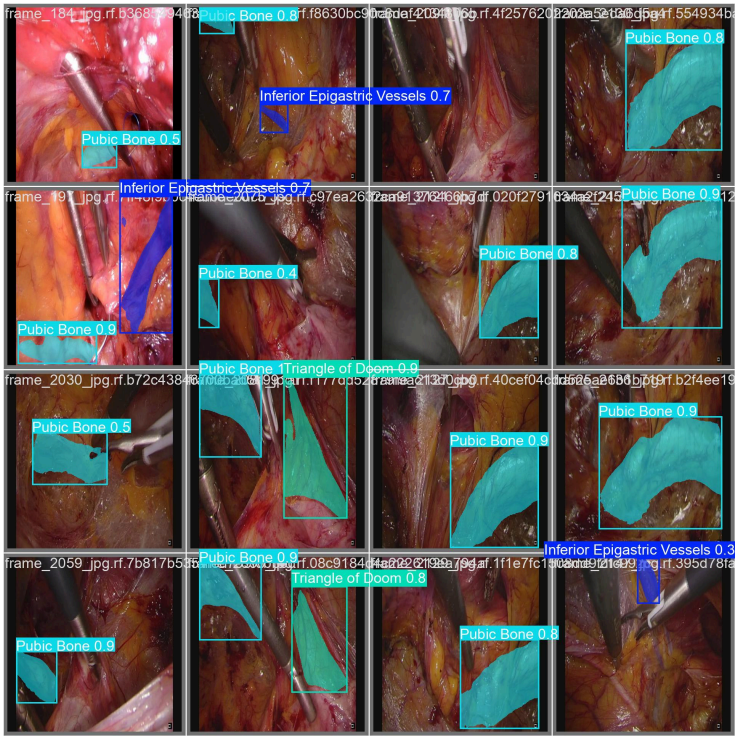
3.2 Results of Model Comparison experiments:

3.2.1 YOLOV11L-SEG model:

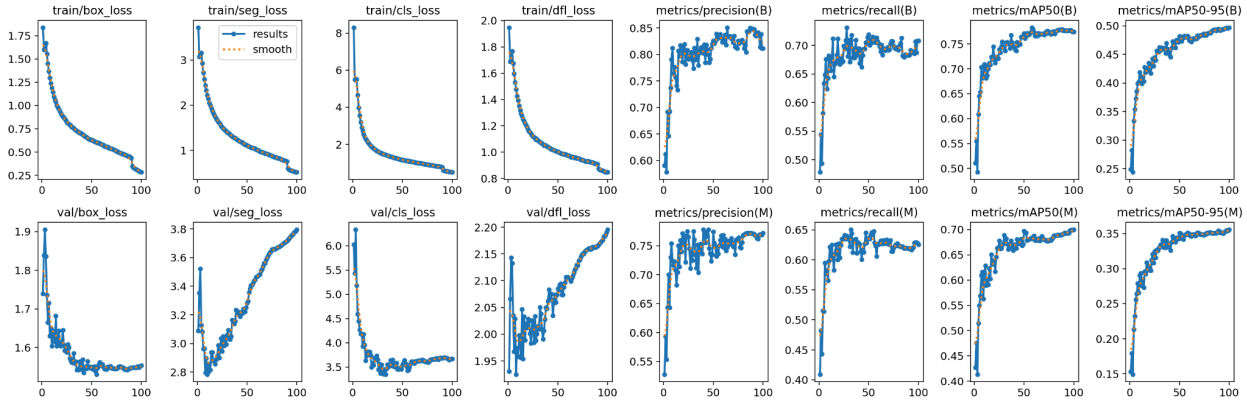
Validation dataset's sample batch Ground Truth labels



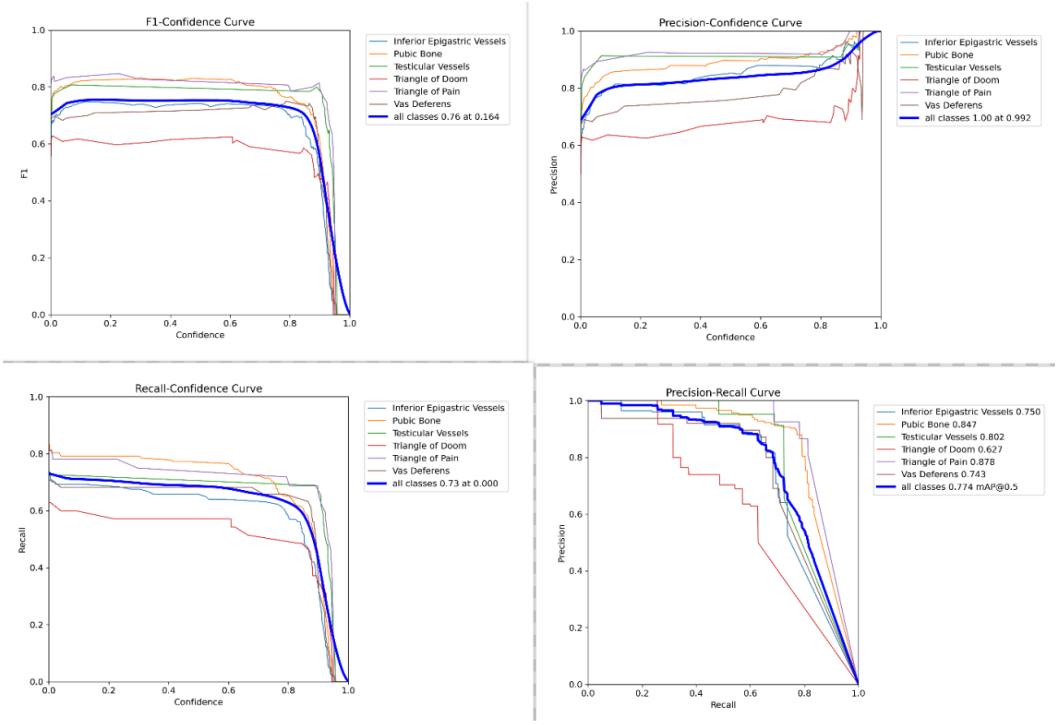
Validation dataset's sample batch predictions from Model Yolov11l-seg:



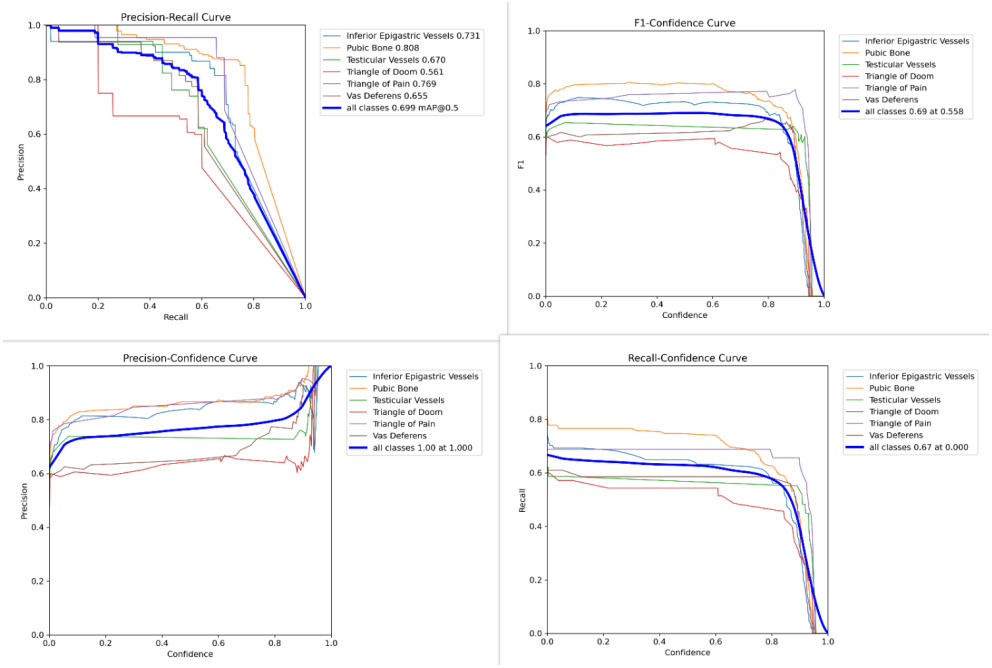
Loss and Performance metrics plotted during training



Bounding Box evaluation metrics Curve



Mask evaluation metrics Curve



Inferences from each plot:

- Box F1- Confidence
  • Peak overall F1 ≈ 0.76 at conf ≈ 0.16 → a low threshold maximizes box F1.
  • Strong classes across most conf: Triangle of Pain, Pubic, Testicular; Triangle of Doom (ToD) lowest.

- Box Precision-Confidence
  • Precision rises with threshold; ≈1.00 at conf ≈ 0.992 (recall is then tiny).
  • High-precision order at mid–high conf: ToP ≥ Pubic ≥ Testicular ≥ IEV ≥ Vas ≫ ToD.

- Box Precision-Recall (PR)
  • Overall mAP@0.5 ≈ 0.774.
  • Per-class mAP: ToP 0.878, Pubic 0.847, Testicular 0.802, IEV 0.750, Vas 0.743, ToD 0.627.

- Box Recall-Confidence
  • Max overall recall ≈ 0.73 at conf ≈ 0; recall steadily drops as conf increases.
  • Pubic / ToP retain recall longer than others; ToD decays earliest.

- Mask F1-Confidence
  • Peak overall F1 ≈ 0.69 at conf ≈ 0.56 → masks prefer a higher threshold than boxes.
  • ToP leads; ToD remains weakest.

- Mask Precision-Confidence
  - Precision increases with threshold; ≈1.00 at conf ≈ 1.00.
  - Pubic / IEV / ToP reach high precision early; ToD lowest.

- Mask Precision-Recall (PR)
  - Overall mAP@0.5 ≈ 0.699 (harder than boxes).
  - Per-class mAP: Pubic 0.808, ToP 0.769, IEV 0.731, Testicular 0.670, Vas 0.655, ToD 0.561.

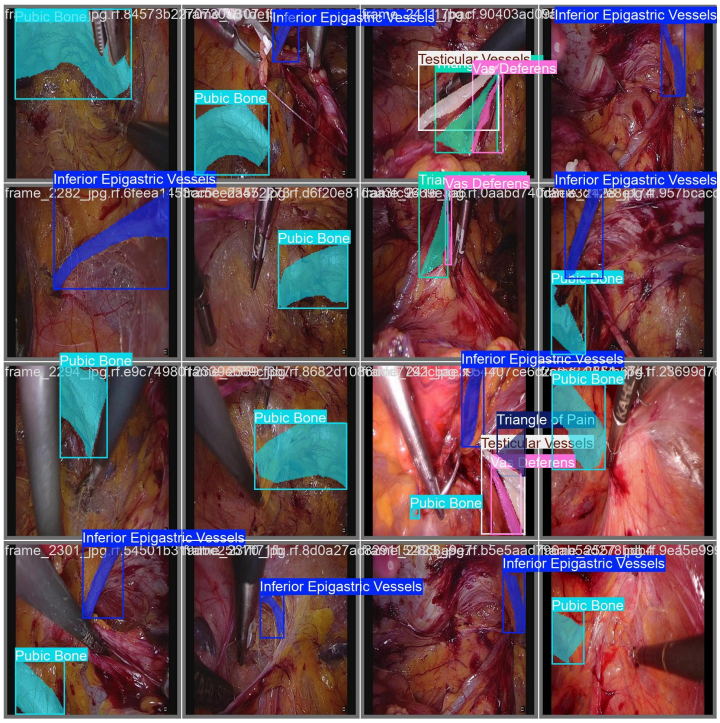
- Mask Recall-Confidence
  - Max overall recall ≈ 0.67 at conf ≈ 0; declines with threshold.
  - ToP / Pubic hold recall longer; ToD drops soonest.

Quick takeaway:

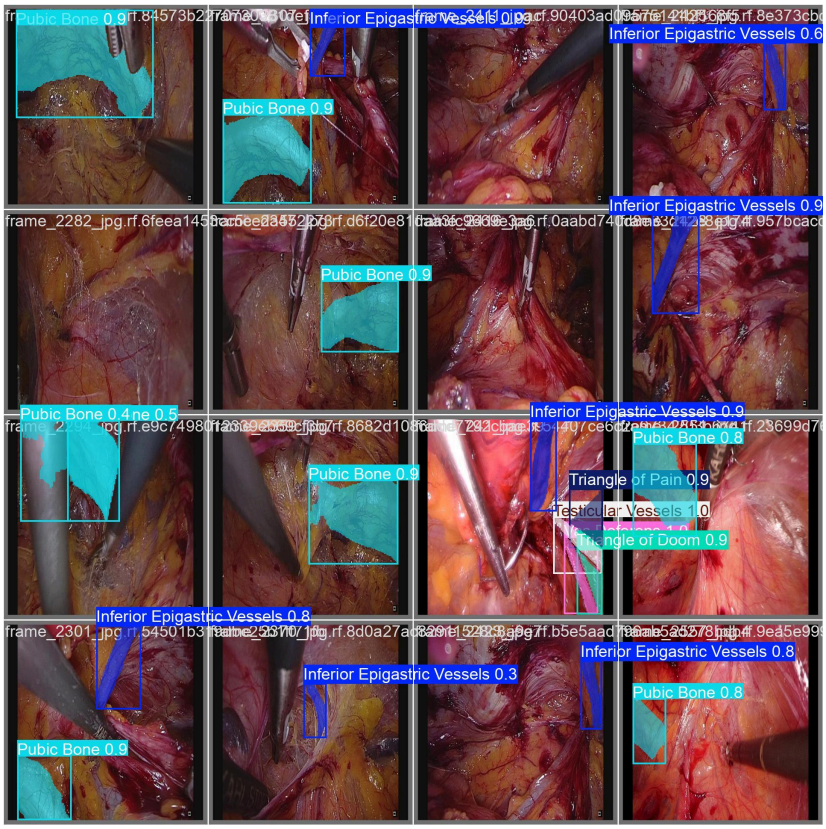
We can use conf ~0.15–0.2 for boxes (best F1) and ~0.55–0.6 for masks. Triangle of Pain and Pubic Bone are consistently strongest; Triangle of Doom is the hardest class.

3.2.2 YOLOV11M-SEG model:

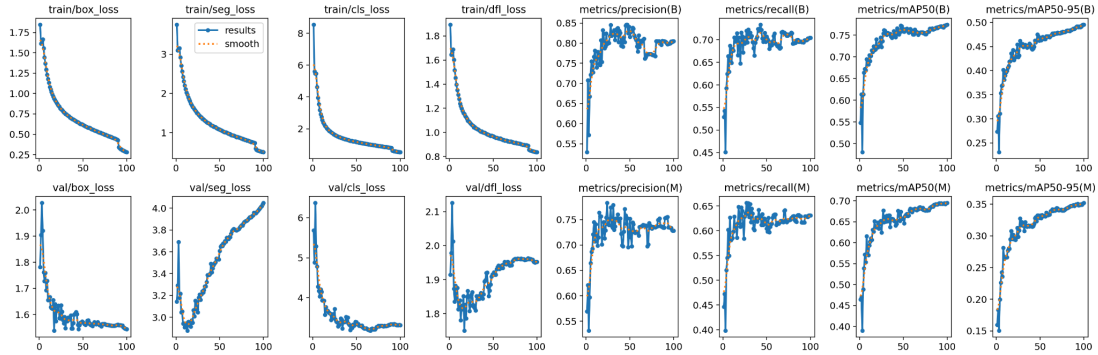
Validation dataset's sample batch Ground Truth labels



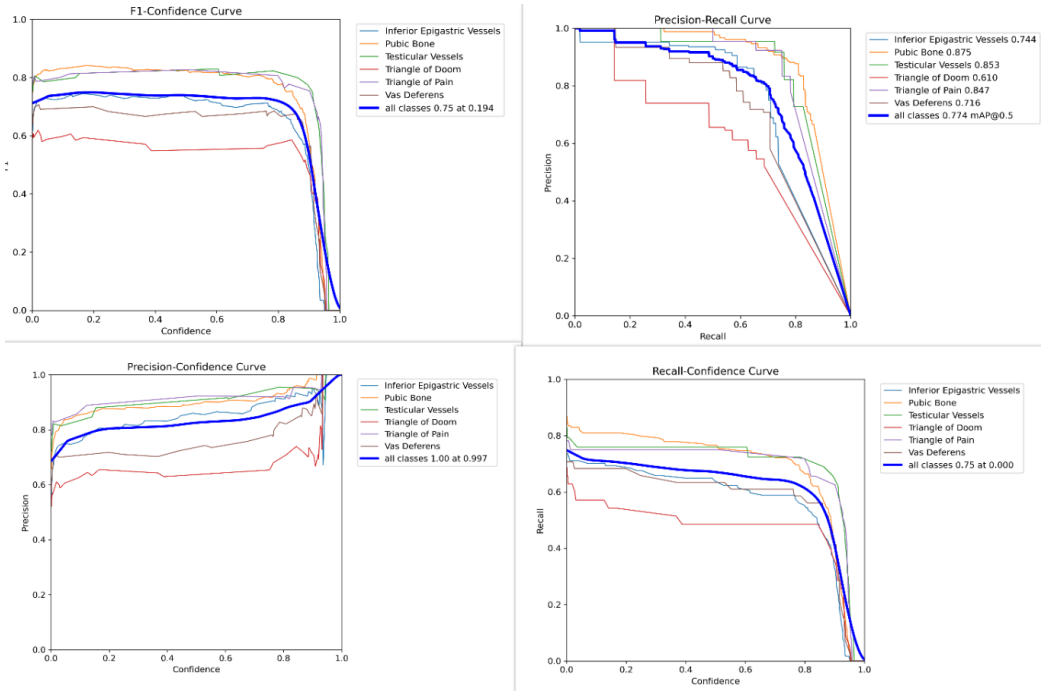
Validation dataset's sample batch predictions from Model Yolov11m-seg:



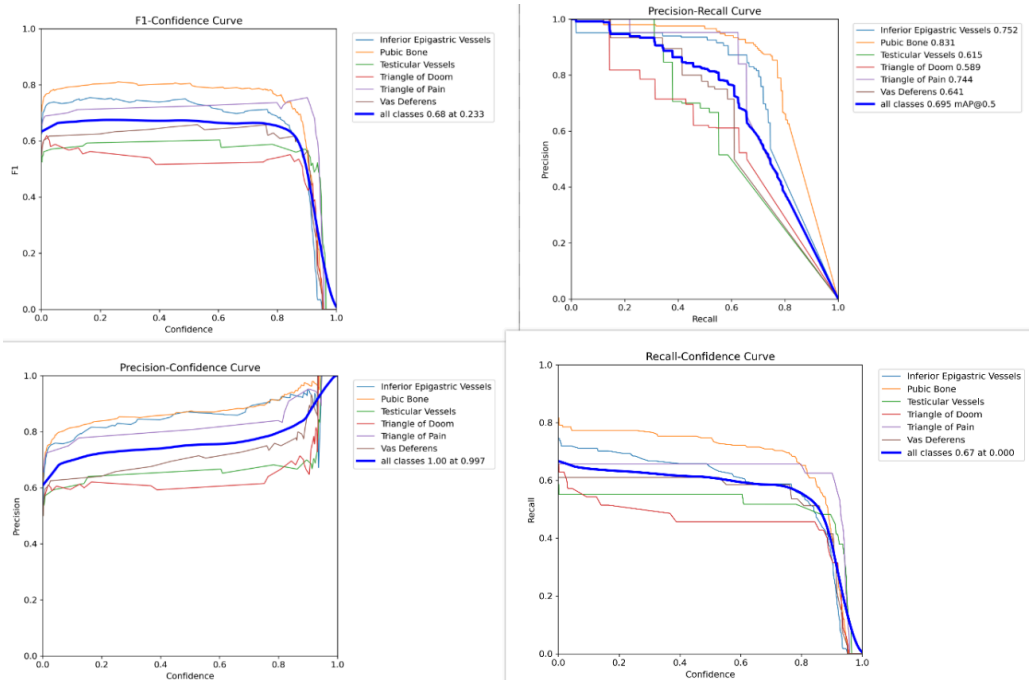
Loss and Performance metrics plotted during training



Bounding Box evaluation metrics Curve



Mask evaluation metrics Curve



Inference from each plot:

- Box F1-Confidence
  • Peak overall F1 ≈ 0.75 at conf ≈ 0.19 → a low threshold gives the best box F1.
  • Per-class: Pubic/Testicular/ToP stay high across mid-confs; “Triangle of Doom” lags.

- Box Precision-Confidence
  • Precision climbs with threshold; hits ~1.0 at conf ≈ 0.997 (but recall will be tiny).
  • Ordering at high conf: Pubic ≳ Testicular ≳ ToP ≳ IEV; ToD is least precise.

- Box Precision-Recall (PR)
  • Overall mAP@0.5 ≈ 0.774.
  • Class ranking (best→worst mAP): Pubic (0.875), Testicular (0.853), Triangle of Pain (0.847), IEV (0.744), Vas (0.716), Triangle of Doom (0.610).

- Box Recall-Confidence
  • Max overall recall ≈ 0.75 at conf ≈ 0.0 → recall drops steadily with threshold.
  • Pubic/Testicular maintain higher recall at mid-confs; ToD falls earliest.

- Mask F1-Confidence
  • Peak overall F1 ≈ 0.68 at conf ≈ 0.23 → slightly higher optimal threshold than boxes.
  • Triangle of Pain leads; ToD remains weakest.

- Mask Precision-Confidence
  • Precision increases with threshold; overall ~1.0 at conf ≈ 0.997.
  • Pubic and ToP reach high precision earlier; ToD lowest.

- Mask Precision-Recall (PR)
  • Overall mAP@0.5 ≈ 0.695 (lower than boxes → masks are harder).
  • Class ranking: Pubic (0.831), IEV (0.752), Triangle of Pain (0.744), Vas (0.641), Testicular (0.615), Triangle of Doom (0.589).

- Mask Recall-Confidence
  • Max overall recall ≈ 0.67 at conf ≈ 0.0; declines with threshold.
  • Pubic retains the best recall; ToD and Testicular drop earliest.

Quick takeaway:

Use a low-mid confidence threshold (~0.2) for best F1; increase it only if you need very high precision and can afford recall loss. Pubic Bone and Triangle of Pain are strongest; Triangle of Doom is consistently the hardest class.

Comparison of metrics and inference:

Comprehensive summary:

Overall (boxes): tie on mAP@0.5 (0.774 each).

Overall (masks): slight edge to *YOLOv11-l-seg* (0.699 vs 0.695).

Best F1 thresholds: boxes favor low conf; masks favor higher conf on *l-seg*.

Detailed comparison for Bounding boxes:

Global curves

- **Peak F1:**
  - **m-seg: ~0.75 @ conf≈0.19**
  - **l-seg: ~0.76 @ conf≈0.16**
- **mAP@0.5: 0.774** (both)

- **Max recall (conf→0):**
    - **m-seg:** ~**0.75** ▸ slightly higher
    - **l-seg:** ~**0.73**
- **Precision–confidence:** both climb to ~**1.0** near **conf≈0.99**

Per-class box mAP@0.5

- **m-seg stronger:** Pubic **0.875>0.847**, Testicular **0.853>0.802**
- **l-seg stronger:** Triangle of Pain **0.878>0.847**, IEV **0.750>0.744**, Vas **0.743>0.716**, Triangle of Doom **0.627>0.610**

Box threshold tip: start around **0.15-0.20** for both; nudge up only if we need more precision.

Detailed comparison for Masks:

Global curves

- **Peak F1:**
    - **m-seg**: ~0.68 @ conf≈0.23
    - **l-seg:** ~0.69 @ conf≈0.56 (slightly higher F1 but needs a much higher threshold)
- **mAP@0.5**:
    - **m-seg**: 0.695
    - **l-seg**: 0.699 (slight win)
- **Max recall (conf→0)**: ~0.67 for both
- **Precision-confidence**: both → ~1.0 near conf≈1.0

Per-class mask mAP@0.5

- **m-seg stronger:** Pubic 0.831>0.808, IEV 0.752>0.731, Triangle of Doom 0.589>0.561

- **l-seg stronger**: Triangle of Pain 0.769>0.744, Vas 0.655>0.641, Testicular 0.670>0.615

Mask threshold tip:

- **m-seg:** start ~0.20-0.25 (balanced F1).

- **l-seg:** start ~0.55-0.60 (its F1 peaks higher at a high threshold).

4. Conclusion and Future scope discussions

The output of our project is a model and a system that can be used to accurately annotate surgical videos and images. This will be used by the stakeholders to help train students and future surgeons when learning about the TEP surgical procedure. However, our working framework allows us to do the same for other surgical procedures as it can be trained to detect any types of anatomical parts on surgical videos. Furthermore, this project also serves as an excellent foundation for further expansion. In this future, the scope can be expanded according to the needs of our stakeholders, such as:

- VLM integration for detecting tool names and their association with anatomical sections through zero-shot prompting and providing good descriptions of tools, a VLM can be integrated into the system to annotate the different surgical instruments that can be seen. Furthermore, we can also integrate VLM to tell where certain tools are meant to be used on different anatomical areas. Thus, we will further explore advanced VLM features for prompt-driven queries (e.g., "highlight the dissecting tool") and assess whether zero/low-shot capabilities can complement supervised anatomy detection, with all outputs continuously validated by clinician feedback to ensure clinical plausibility and high instructional value

- Real-time segmentation of anatomical parts beyond teaching, support of real-time inference can also be beneficial to live surgery by providing clear outlines of critical anatomical parts. This would help speed up surgery as surgeons would spend less time identifying parts and reduce risk of mistakes such as putting sutures on major vessels during the procedure.

- It is also possible to calculate distances between tools against objects. However this is a major challenge if it were to be implemented as it requires at least two cameras for depth perception. Which is very uncommon, as surgeons would commonly work with both hands and only use one camera in one hand.