lajellu@my.yorku.ca
212319356
Laily
Ajellu

dbadiei@my.yorku.ca
219399328
David
Badiei

tran03@my.yorku.ca
220168829
Tran
Tran

andrewk1@my.yorku.ca
219713213
Andrew
Kolosowski

manstein@my.yorku.ca
218748947
John
Xie

Computer Vision-Based Gastric Pathway Verification for Reducing Anesthesia Risk

York University, Lassonde School of Engineering

EECS 3404: Applied Machine Learning

Dr. Navid Mohaghegh

1: Problem

1.2 Problem Definition

Task
This project frames the problem as segmentation of the five relevant anatomical structures — antrum, liver, aorta, pancreas, and superior mesenteric artery (SMA) — with labeled outlines overlaid on the image to help the reader visually locate the antrum quickly [1], paired with classification of gastric content state into three segmentation classes: Empty, Liquid, and Solid. The segmentation output directly supports the I-AIM "Interpretation" step by making the antrum and its surrounding landmarks (liver anterior, pancreas posterior, aorta deep) immediately visible, rather than requiring the reader to mentally locate them the way an unassisted sonographer currently must [1].

Input
Raw gastric-ultrasound video sequences, annotated with clinical acquisition posture (Supine or RLD) [1], paired with ground-truth content-state assignments (Baseline, Water, or Solid) established during controlled experimental conditions.

Output

1. Segmented, labeled overlay of the five anatomical structures (antrum, liver, aorta, pancreas, SMA) on the input frame/video.
2. Automated classification of gastric content into three segmentation categories: Empty, Liquid, or Solid, mapped from the initial experimental states (Baseline → Empty, Water → Liquid, Solid → Solid).

Primary success metric
Quantification of segmentation performance across the multi-class dataset, specifically evaluating mask precision, mask recall, mask mAP50, and mask mAP50–95.

Scope boundary: This development cycle is restricted to the anatomical presentation of normal adult subjects.

Assumptions and scope. The methodology assumes that experimental condition labels accurately reflect the physiological state of the stomach during acquisition and that human-reviewed CVAT annotations serve as a reliable gold standard for model training. It is further assumed that input ultrasound imagery contains the requisite anatomical clarity for interpretation. The system is presented as a research-oriented annotation-assistance framework; consequently, these results should be viewed within the context of a controlled development workflow rather than as evidence of broader clinical generalization.

1.2 Problem Motivation

Pulmonary aspiration under anesthesia carries significant morbidity and mortality, and is the leading cause of death from anesthesia-related airway events [1]. Even in fasted elective-surgery patients, baseline risk is approximately 1 in 4,000 and inaccurate risk assessment is a frequent root cause [1]. The dangerous failure mode is a "latent full stomach": a patient who followed fasting instructions but whose stomach isn't actually empty, historically judged only by patient history, which fails exactly when it's most unreliable.

Problem 1: Fixed fasting timelines don't reflect actual patient risk.

Standard ASA fasting guidelines (2h clear fluids / 6h light meal / 8h fatty food) assume a fixed timeline is sufficient for gastric emptying. But several common comorbidities and physiologic conditions, diabetic gastroparesis, achalasia, advanced renal or hepatic dysfunction, and critical illness among them, can prolong gastric emptying despite adequate fasting [1]. Diagnosis category alone does not reliably predict actual gastric content; only direct measurement does. This matters more because not all classification errors carry equal clinical risk: a false "empty" reading is more dangerous than a false "full" reading, since it can lead to standard induction under real aspiration risk, whereas a false "full" reading only causes unnecessary precautions. This is why the negative predictive value of gastric ultrasound, its ability to correctly rule out a full stomach, is considered its most clinically important diagnostic property [1]. A risk-assessment tool needs to reflect this asymmetry directly, not approximate it through blanket time-based rules.

Problem 2: Late cancellations are costly, and better assessment enables earlier decisions.

A surgery room booking cancelled less than 2 hours in advance results in substantial extra cost to the hospital, compared to a

cancellation made 6 hours in advance, which allows resources to be reallocated in time. This isn't just a hypothetical efficiency gain: a prospective study of 38 elective surgical patients who had not complied with fasting instructions found that adding point-of-care gastric ultrasound changed anesthetic management in 72% of cases compared to relying on history alone, with a trend toward fewer surgical delays [1]. Faster, more confident risk assessment doesn't just improve safety, it directly supports earlier, better-informed scheduling decisions.

Problem 3: Clinicians are experiencing app fatigue and alert fatigue, so automation has to reduce cognitive burden, not add to it.

Gastric POCUS interpretation is operator-dependent: on average, approximately 33 supervised practice examinations are needed for anesthesia fellows to reach 95% diagnostic accuracy, and even experienced sonographers find up to 3–5% of scans inconclusive [1]. In urgent, ICU, or resource-limited settings, the clinician available may not yet have that experience curve behind them, and image quality is often degraded by patient positioning constraints. The goal isn't to hand clinicians another tool to check, it's to close the experience gap in exactly the moments where waiting for a specialist isn't an option, without adding another alert to an already-saturated workflow.

Problem 4: GLP-1 receptor agonists have created a large, growing population where standard guidance is actively shifting:

GLP-1 receptor agonists (semaglutide, tirzepatide) are prescribed at massive scale for diabetes and weight loss, and mechanistically delay gastric emptying. A documented case of large-volume regurgitation in a fasted, non-diabetic, non-obese semaglutide patient after roughly 20 hours of fasting triggered a wave of similar reports, and multisociety guidance now recommends extending solid fasting to 24 hours for this population [2]. The one constant across every revision of that guidance, across multiple countries, is the recommendation to use gastric ultrasound for risk stratification when in doubt.

2: Related Works

Recent work has approached automated gastric-ultrasound analysis from several directions, but no single system identified in this review combines the complete workflow developed in this project: **raw ultrasound video → physiologically informed frame selection → AI-assisted gastric-content and anatomical annotation → CVAT human review → reviewed-dataset construction → model training and redeployment**.

Segmentation-focused approaches. Zou et al. developed a Swin-UNet-based system for real-time segmentation of the gastric antrum from bedside ultrasound video, also localizing the liver and aorta as anatomical landmarks, achieving a mean Dice coefficient of 87.36% and strong agreement with experienced clinicians (ICC 0.813) [3]. Their goal was tracking antrum motility rhythm over time for ICU feeding-tolerance monitoring, not aspiration risk grading, the system performs no Perlas classification and does not segment the pancreas, which is different from our project where we also segment the pancreas.

Volume-and-grade prediction approaches. Liu et al. developed eight machine learning models (Random Forest, XGBoost, AdaBoost, SVM, and others) to predict gastric volume from age, Perlas grade, and RLD-CSA, substantially outperforming the classical Perlas linear formula, with AUCs up to 0.96 for detecting high gastric volume [4]. However, this approach starts from an already human-measured CSA and grade, it contains no image segmentation step, and requires age as an input feature, which is unavailable in our dataset.

Non-imaging clinical prediction approaches. Jin et al. built a Random Forest model to predict full-stomach status from clinical variables alone (age, diabetes, GERD, fasting time, diet type), explicitly targeting settings where POCUS is unavailable (AUROC 0.837) [5]. While their POCUS-augmented variant performed better, their core contribution is a non-imaging fallback tool, which is a different use case from an image-driven pipeline and not directly comparable to this project's approach.

The gap. Across all three studies, no system performs the complete pipeline this project targets: segmenting and labeling the antrum alongside the liver, aorta, pancreas, and SMA directly from raw ultrasound video, and using that segmentation to drive both Perlas grade and content-state classification in one integrated tool. Zou et al. solves segmentation but stops before clinical grading and omits the pancreas [3]; Liu et al. solves grading-to-volume but requires pre-measured inputs a clinician must already have extracted [4]; Jin et al. sidesteps imaging entirely [5]. None of the three account for GLP-1 receptor agonist status, despite it being flagged as a rapidly growing risk factor in current clinical guidance [2]. This project's multi-task segmentation-plus-classification design, built directly on labeled ultrasound video, targets exactly this integration gap.

3: Datasets

3.1 Dataset Description

3.1.1 Antrum detection subset.

To train and evaluate our machine learning models for antral gastric pathway verification, we compiled a multi-source dataset comprising ultrasound imagery captured across various physiological states and body positions. The dataset combines newly acquired clinical data with historical repository data to ensure model robustness and generalization.

The primary data source consists of prospective ultrasound recordings collected directly from human volunteers in the Dr Perlas Study group under controlled experimental conditions. Each participant underwent ultrasound imaging across three distinct physiological states: baseline (fasting or empty stomach), liquid state (post-ingestion of liquid), and solid state (post-ingestion of solid food). For every physiological state, recordings were captured in two standard clinical postures: Right Lateral Decubitus (RLD) and Supine. The ground truth labels for the current state and posture were systematically embedded into the video file naming conventions. Individual image frames were extracted from these video recordings, and the spatial boundary of the gastric antrum in each frame was manually annotated by team members using the Computer Vision Annotation Tool (CVAT). These annotations were exported as text files containing coordinate segments outlining the antral region.

The secondary data source consists of an inherited benchmark dataset from previous researchers working on the repository. This dataset includes pre-extracted static ultrasound images paired with separate text files containing state classifications and corresponding antrum segmentation boundaries annotated by the prior team. To enhance dataset diversity and evaluate cross-pipeline generalization, samples from this secondary source were partitioned and integrated into both the training and validation sets alongside the newly acquired volunteer data.

3.1.2 SupportingStructure development subset.

The SupportingStructure subsystem was developed and evaluated using gastric-ultrasound frames already available within the project's CVAT environment. Development did not treat all frames as equivalent evidence. Early exploratory work used Tasks 1, 2, and 5 to evaluate candidate behavior and identify failure modes. Tasks 31 and 33 were later used for live CVAT workflow validation, while Task 32 was intentionally retained as an inconclusive case because the anatomy was not sufficiently clear for a confident liver validation.

Task 34 was then created as the permanent systematic validation and regression reference for the final SupportingStructure methodology. It contains five consecutive frames corresponding to Task 1 frames 0–4. Each validation frame contains an abdominal-wall, liver, pancreas, and post-generation AntrumContext object, giving 20 final persisted validation shapes.Task 34 should be treated as a validation/regression reference rather than an independent held-out test set because its frames originate from Task 1 and are temporally adjacent. Broader generalization therefore still requires evaluation on materially different patients, scans, probe positions, or acquisition conditions.

3.2: Dataset preprocessing

Project preprocessing begins with frame selection and annotation preparation. The existing frame-selection pipeline supports quality-only selection, contraction-aware selection, and a fallback path when no usable peristaltic signal is detected. This reduces the chance that low-quality or contraction-distorted frames silently enter the annotation/training workflow.Annotation protocol development was informed by clinical requirements gathered through consultation with Dr. Anahi Perlas and Dr. Sena Aflaki; the detailed role of these consultations is described in Section 4.

As the original ultrasound video from our collaborator contains the state of the antrum which should not be known to the model, so we manually masked the portion to avoid the data leakage.This preprocessing has applied to gold_label dataset and review_gold_label dataset.

For SupportingStructure generation, selected ultrasound frames are processed with image-derived fan geometry and role-specific candidate construction. The abdominal wall is constrained by a smoothed fan envelope; the final pancreas method uses Otsu thresholding followed by morphological processing, connected-component selection, contour extraction, and simplification; and the final liver method combines a broad human-specified topology with image-derived interface tracing. Polygon simplicity, fan containment/attachment, overlap, and duplicate checks are evaluated before persistence.

AntrumContext is not part of preprocessing or candidate generation. It is copied point-for-point only after SupportingStructure generation and approval, under a separate context-only label, so existing gastric-content/antrum geometry cannot leak into the generation step.

4: Methodology

The project methodology was informed by direct consultation with clinical domain experts to establish the anatomical and procedural requirements for interpreting gastric ultrasound and producing meaningful annotations. In particular, meetings with Dr. Anahi Perlas were used to clarify the clinically relevant anatomy, ultrasound appearance of gastric contents, and appropriate interpretation of surrounding anatomical landmarks. Additional discussions with Dr. Sena Aflaki also contributed to the team's understanding of ultrasound interpretation and the practical requirements of the proposed workflow.

Dr. Perlas's guidance was treated as the primary source for annotation clarification because of her direct expertise in gastric point-of-care ultrasound and her involvement with the dataset. These discussions helped the team translate clinical ultrasound interpretation into concrete annotation requirements before implementing the computer-vision workflow.

For example, Dr. Perlas clarified that the pancreas appears relatively echogenic and lies above the aorta, while the aorta appears dark because it contains blood, and the superior mesenteric artery appears as a dark structure above the pancreas. She also clarified characteristic gastric-content appearances: fluid generally appears dark because it transmits ultrasound waves well, whereas solid gastric content tends to appear white or grey toward the anterior surface because of mixed air and may produce posterior blurring or acoustic artifacts. These clarifications were used to guide anatomical interpretation during annotation and human review rather than being treated as independent model inputs.

The consultations therefore served as a requirements-elicitation and annotation-protocol step within the ML workflow: clinical knowledge was first translated into observable image characteristics and spatial relationships, which then informed manual annotation, SupportingStructure candidate review, and subsequent technical validation.

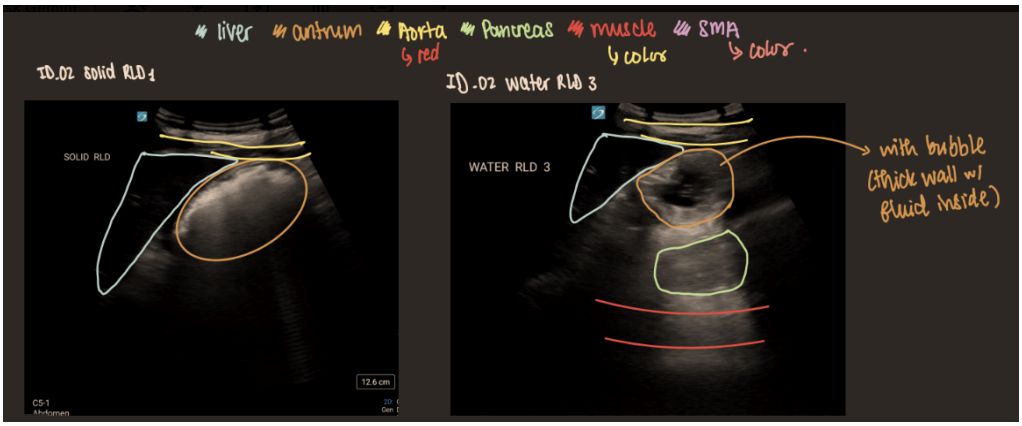


Figure 1. Examples used during domain-expert clarification of gastric-ultrasound anatomy and content appearance. Clinical consultation informed interpretation of structures including the antrum, pancreas, aorta, SMA and liver, as well as characteristic appearances of empty, liquid and solid gastric states. These observations informed annotation requirements and human review rather than serving as direct model inputs.

4.1: Model architecture

Development began from an existing human-in-the-loop gastric-ultrasound annotation and training pipeline. The inherited system extracted frames from source videos, generated gastric-content preannotations using a YOLO seed detector with optional SAM refinement, enforced filename-derived content labels, presented the resulting masks in CVAT for manual review, exported corrected annotations, and merged those annotations into a YOLO training dataset. Reviewed data could then be used to train a larger YOLO26x segmentation model and redeploy the promoted model to CVAT through Nuclio-assisted automatic annotation.

At a high level, the inherited workflow connected video preprocessing, machine-learning-assisted pre-annotation, human review in CVAT, reviewed-dataset construction, model training, and deployment back into CVAT:

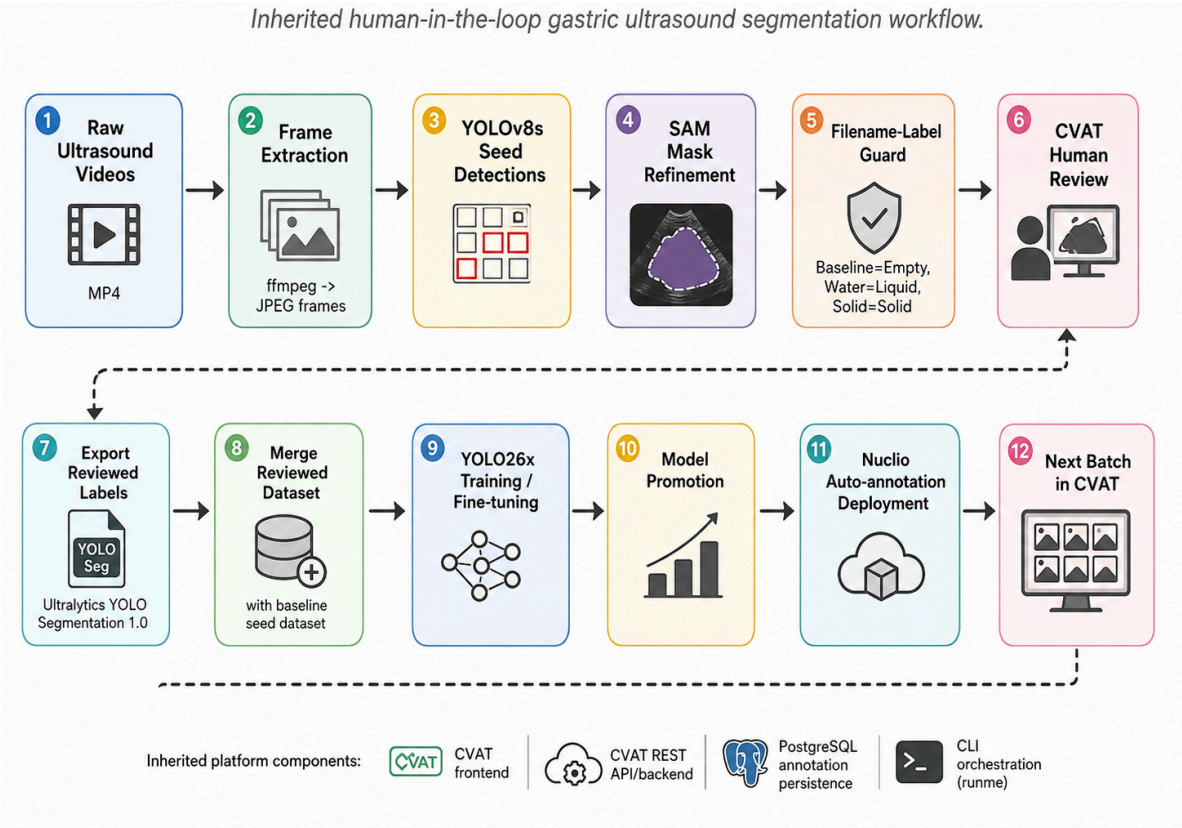


Figure 2: Inherited human-in-the-loop gastric-ultrasound segmentation workflow, including frame extraction, YOLOv8s/SAM pre-annotation, filename-label enforcement, CVAT review, reviewed-dataset construction, YOLO26x fine-tuning, model promotion, and Nuclio-assisted auto-annotation.

Video extraction and dataset preparation

Source gastric-ultrasound videos are decoded into image frames using `ffmpeg`. The extraction pipeline records frame- and video-level metadata and preserves the relationship between individual frames and their source videos. Video filenames also encode the controlled gastric-content condition used by the existing workflow: Baseline, Water, or Solid.

Gastric-content pre-annotation

The inherited annotation pipeline supports three pre-annotation modes: no machine-generated labels, YOLO-only labels, or YOLO-guided SAM segmentation. The default gastric pre-annotation workflow uses an existing YOLOv8s seed model to propose gastric-content detections and Segment Anything Model (SAM) to refine the proposed regions into polygon masks.

A filename-label consistency guard is then applied before annotations proceed to CVAT. The generated polygon geometry is retained, while the class assignment is constrained to match the known experimental condition encoded in the source-video filename:

- Baseline → Empty
- Water → Liquid
- Solid → Solid

This provides weak supervision for gastric-content state while allowing the segmentation models to determine spatial mask geometry.

CVAT annotation platform

CVAT provides the human-review layer of the inherited system. The repository maintains a pinned checkout of the CVAT source and launches the application through Docker Compose. The standard CVAT frontend provides interactive frame navigation and annotation editing, while its backend and database services persist in the annotation state.

The project communicates with CVAT programmatically through the CVAT Python SDK and CVAT's built-in REST API rather than through a separate project-specific CRUD API. This allows the pipeline to create annotation resources, upload images and prelabels, retrieve annotations, and export corrected labels while retaining the standard CVAT user interface for manual review.

In the original inherited organization, one CVAT Project grouped a dataset, each source video was represented as a separate CVAT Task, and the frames belonging to that video were exposed through the associated Job for review. This hierarchy formed part of the engineering baseline and was later simplified as part of the project's infrastructure optimization.

Human review and persistent annotation state

Within CVAT, reviewers inspect the machine-generated gastric-content masks, correct polygon boundaries, remove false-positive detections, and add missed regions. These edits are stored by CVAT's persistent annotation backend. Human-reviewed empty annotations are also meaningful because deletion of an incorrect prelabel prevents that candidate from entering the subsequent reviewed training dataset.

Reviewed-dataset construction and model training

Reviewed annotations can be exported from CVAT in Ultralytics YOLO segmentation format and merged with the existing seed dataset. The inherited training pipeline uses `train_yolo_seg.py` as a reproducible wrapper around Ultralytics training.

Two YOLO model roles should be distinguished within the architecture. The smaller pretrained gastric YOLOv8s model serves as the existing seed detector used for pre-annotation, whereas the default model used for subsequent reviewed-data training is YOLO26x segmentation. Model checkpoints and training outputs are maintained separately under the repository's model and run directories.

Gold_label Dataset: The ground-truth dataset consists of human-annotated labels curated and verified by team members. The dataset contains 154 total instances across three classes: 66 in the water state, 66 in the solid state, and 22 in the empty state. To ensure proportional class representation across splits, the data was partitioned into an 80/20 train-validation ratio. Given the lower frequency of the empty state class, 11 instances were assigned to the validation set alongside 11 instances from each of the other two classes (yielding 55 water, 55 solid, and 11 empty instances in the training set).

Gold_label_Review_dataset: To address class imbalance in the original Gold_label training set, the **121 training frames** were combined with **3,179 additional frames** from an inherited reviewed-data pool, producing an expanded training set of 3,300 instances: **1,013 Empty, 1,259 Liquid, and 1,028 Solid**. The held-out Gold_label validation data were kept separate from this expanded training pool to preserve the evaluation split. However, the provenance and verification history of portions of the inherited supplementary dataset could not be fully reconstructed from the available project records. This is therefore treated as a **dataset-provenance limitation**, and results obtained from models trained on the expanded dataset should not be interpreted as equivalent to results obtained using only the newly human-reviewed Gold_label dataset.

Model deployment and feedback loop

Promoted gastric-segmentation checkpoints can be deployed to CVAT through a Nuclio serverless function. The Nuclio integration provides a dedicated inference endpoint that accepts image data, executes the deployed YOLO segmentation model, and returns polygon predictions in the format expected by CVAT's automatic-annotation workflow.

This creates a feedback loop in which reviewed annotations can be incorporated into a new training dataset, used to train an improved gastric-content model, and subsequently redeployed to assist annotation of later data.

Command-line orchestration

The repository-level `runme` command acts as the primary orchestration interface. It coordinates the individual modules responsible for environment setup, video extraction, pre-annotation, CVAT deployment and upload, reviewed-data export and merging, training, evaluation, model promotion, and automatic-annotation deployment.

The inherited architecture therefore provided the project's foundational **video → machine pre-annotation → human review → training → deployment** workflow. Subsequent project contributions extended this baseline through frame-selection improvements, CVAT infrastructure optimization, annotation-preview improvements, and the SupportingStructure subsystem; these modifications are described separately so that inherited functionality is not conflated with work completed during this project.

4.1.1 Final Integrated Project Architecture

Figure 3 shows the final integrated system architecture produced by this project. The architecture combines the inherited gastric-content annotation pipeline with the project's main engineering and methodological contributions, while preserving the existing reviewed-data training and deployment loop. At a high level, the system begins with raw gastric-ultrasound videos, performs frame extraction and AI-assisted pre-annotation, supports human review in CVAT, and feeds reviewed annotations back into model training and deployment. The final architecture therefore retains the original human-in-the-loop segmentation workflow while extending it with more structured frame selection, streamlined CVAT integration, and a separate safety-controlled SupportingStructure workflow.

The first major addition is the **frame-selection subsystem**, which reduces the number of frames forwarded to downstream review by prioritizing frames that are both visually usable and physiologically appropriate. Candidate frames are first scored using image-quality measures. When a usable antral-area or peristaltic signal is available, contraction-aware selection is applied to identify relaxation peaks corresponding to frames near the maximum antral cross-sectional area between contractions. When no reliable signal is available, the system falls back to quality-only selection. In both cases, the output is a selected-frame manifest that records the frames chosen for downstream processing and annotation.

The second major architectural change is the **optimized CVAT integration**. In the final system, CVAT remains the primary human-review platform, but its configuration is streamlined for the gastric-ultrasound workflow. The platform uses one CVAT Task per source video, removes the unnecessary Project grouping layer from the uploader workflow, and supports direct-disk frame serving for image tasks. The CVAT review interface was also extended with a **read-only Smooth Preview that applies polygon smoothing for visualization without modifying the stored annotation geometry**. Database-backed annotation state, the React-based review interface, and standard CVAT API functionality are retained for persistent human annotation and export. This preserves the essential human-in-the-loop review capability while reducing unnecessary infrastructure complexity.

The third major addition is the **SupportingStructure branch**, which introduces a separate human-supervised workflow for supporting anatomical structures. This branch is intentionally designed so that candidate generation and persistence are separated. SupportingStructure candidates are first produced in a dry-run, evaluated through geometry and topology diagnostics, reviewed by a human, frozen in an approved report, and then authorized for CREATE-only persistence through SHA-256 binding. Independent readback and duplicate/idempotence checks verify the applied result. This design ensures that supporting-structure annotations are added in a controlled, reproducible, and auditable manner.

After human review and reviewed-label export, the integrated architecture returns to the inherited training and deployment loop. Reviewed annotations are merged into the dataset, used for YOLO26x fine-tuning, promoted into a stable model artifact, and deployed back into CVAT through Nuclio for subsequent automatic annotation. The final integrated architecture therefore represents a complete end-to-end system that unifies inherited gastric-content segmentation, project-specific frame-selection improvements, optimized annotation infrastructure, and a safety-controlled SupportingStructure workflow within a single continuous improvement pipeline.

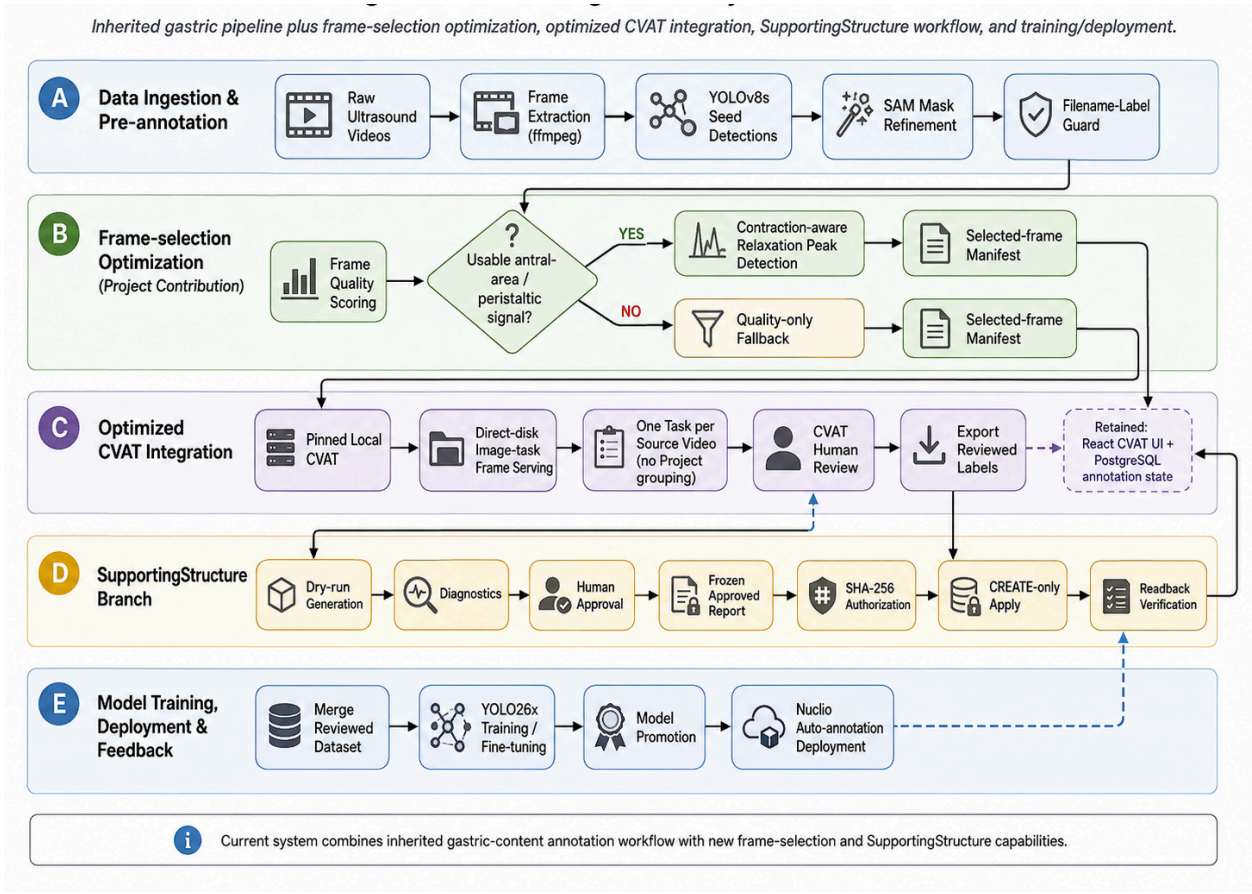


Figure 3. Final Integrated Project Architecture.

Final system architecture combining the inherited gastric-content annotation pipeline with project contributions in frame-selection optimization, streamlined CVAT integration, the SupportingStructure human-in-the-loop workflow, and the existing reviewed-data training, model-promotion, and Nuclio deployment loop.

4.1.2 CVAT Architecture, Infrastructure, and Frame-Selection Optimization

The inherited CVAT-based architecture was reviewed to identify components that introduced unnecessary infrastructure or organizational complexity for the gastric-ultrasound workflow. Two targeted architectural changes were implemented. First, cache-backed frame access for image tasks was replaced with direct-disk frame reads, reducing the infrastructure required for frame delivery while preserving CVAT's interactive review workflow. Second, the CVAT organizational hierarchy was simplified by removing the Project grouping layer and using one CVAT Task per source video, while retaining the Task and Job abstractions required for annotation and manual review.

Components required for persistent annotation, interactive review, and frame-by-frame navigation were deliberately retained rather than removed. The resulting changes therefore focused on reducing unnecessary infrastructure without eliminating CVAT functionality required by the human-in-the-loop workflow.

Existing component	Original purpose	Assessment	Project decision	Result
Frame caching / chunk preloading	Smooth scrubbing	Useful but unnecessarily heavy for image tasks	Replaced cache - backed image access with direct disk reads	Reduced the CVAT deployment from 18 to 16 containers by removing Kvrocks and **`cvat_worker_chunks`**, while preserving comparable manual scrubbing responsiveness.
Project / Task / Job hierarchy	Dataset organization	Project grouping unnecessary for this workflow; Task/Job still required by CVAT	Simplified to one Task per source video	Reduced organizational complexity while preserving CVAT's required Task/Job structure
Database annotation state	Persistent human review	Essential	Retained	Preserved collaborative/manual review
React CVAT UI	Human annotation	Essential	Retained	Maintains interactive review
Tracking / interpolation	Cross-frame tracking	Workflow-dependent	Optional	Can be disabled if unused

Frame selection was optimized separately from these infrastructure changes. The selection pipeline supports **contraction-aware selection when a usable peristaltic signal is available and quality-based selection when physiological timing cannot be reliably determined**. Quality scoring reduces the likelihood that low-quality frames enter the annotation workflow, while

contraction-aware selection uses changes in antral cross-sectional area to identify relaxation periods between contractions. Quality-based selection also provides the fallback path for videos in which no reliable peristaltic signal can be identified.

The implementation specifically evaluates cross-sectional-area measurements, composite image-quality scores, and relaxation-peak detection. When a usable contraction pattern is present, detected relaxation peaks are used to favour frames near the maximum measurable antral cross-sectional area between contractions. When no reliable peristaltic signal is detected, the pipeline falls back to image-quality criteria rather than inferring a physiological signal from noise. These selection paths are exercised through end-to-end integration tests.

Optimization therefore addressed two different sources of pipeline inefficiency: **system-level overhead in the inherited CVAT architecture** and **selection-level inefficiency in determining which ultrasound frames should proceed to annotation and downstream training**. Benchmarking and validation were used to evaluate both infrastructure-level resource usage and annotator-facing responsiveness. Quantitative resource measurements were complemented by standardized manual scrubbing tests so that infrastructure reductions were not treated as successful if they introduced a noticeable degradation to the annotation workflow.

CVAT Frame-Serving Optimization: Verification and Benchmarking

Under the original frame-serving architecture, image chunks were generated through a dedicated chunk worker and cached using Kvrocks. The optimized architecture replaced this workflow with direct-disk chunk serving, in which the required JPEG frames are read from disk and assembled on demand. This allowed two infrastructure services to be removed: **cvat_redis_ondisk** (Kvrocks) and **cvat_worker_chunks**.

Reproduction testing of PR #9 as submitted identified a task-level chunk-routing defect that caused direct task-level chunk requests to fall back to the removed legacy cache path, producing approximately 51-second waits followed by HTTP 429 responses. The defect was isolated, corrected with a minimal routing fix, and covered by eight automated regression tests before the final optimized architecture was benchmarked.

Further tracing confirmed that the defect did not affect normal interactive annotation because the CVAT UI retrieves image chunks through the job-level route, which bypassed the affected task-level path.

Following repair and validation, the final optimized deployment retained the direct-disk architecture and reduced the CVAT stack from **18 to 16 running containers**.

replaces

"The CVAT infrastructure was evaluated to determine whether the existing image-frame caching architecture could be simplified without reducing annotation usability. The optimization replaced the cached image-chunk serving workflow with direct-disk frame access for image-based tasks.

Reproduction testing of PR #9 as submitted identified a task-level chunk-routing defect that caused direct task-level chunk requests to fall back to the removed legacy cache path, producing approximately 51-second waits followed by HTTP 429 responses. The defect was isolated, corrected with a minimal routing fix, and covered by eight automated regression tests before the final optimized architecture was benchmarked.

Further tracing confirmed that the defect did not affect normal interactive annotation, because the CVAT UI retrieves image chunks through the job-level route, which bypassed the affected task-level path.

Under the original architecture, image chunks were generated through a dedicated chunk worker and cached using Kvrocks. The optimized architecture introduced direct-disk chunk serving, in which the required JPEG frames are read from disk and assembled on demand. This allowed two infrastructure services to be removed: **`cvat_redis_ondisk`** (Kvrocks) and **`cvat_worker_chunks`**.

As a result, the CVAT deployment was reduced from **18 to 16 running containers**. The recovered July benchmark also measured an approximately **15.5% reduction in idle memory usage**, while manual frame scrubbing remained responsive."

CVAT Optimization Validation Original vs Final

Metric	Original architecture (`main`)	Final optimized architecture (PR #9 + repair)	Outcome
Running containers	18	16	2 services removed (−11.1%)
Kvrocks	Present	Removed	Reduced infrastructure
cvat_worker_chunks	Present	Removed	Reduced infrastructure
Idle RAM	~3.30 GiB	~2.92 GiB	Reduced by ~11.6%
Peak scrub RAM	3.50 GiB	3.11 GiB	Reduced by ~11.1%
Task-level chunk request	Functional under legacy architecture	HTTP 200 / ~0.34 s	Functional after optimization
Normal UI chunk requests	HTTP 200	HTTP 200	No regression
Manual scrubbing	Smooth enough for normal use	Comparable; brief stalls only during very aggressive scrubbing	No noticeable degradation
Automated regression tests	—	8/8 passed	Final deployment validated

Note: Idle RAM represents the summed memory usage of all running CVAT containers after the standardized post-startup settling period. CPU results are omitted because single-run measurements showed sufficient variability to be treated as directional rather than conclusive.

The recovered July benchmark and the contemporaneous validation run were conducted in separate sessions under different machine states; consequently, the observed idle-RAM reduction differed slightly (~15.5% in July versus ~11.6% in the same-day final comparison). Both measurements nevertheless showed the same direction of change: lower memory usage after removal of Kvrocks and cvat_worker_chunks.

Verification of Interactive Annotation Performance

The optimization was subsequently reconstructed and retested using the same CVAT image task and a standardized manual scrubbing procedure. The test included forward and backward navigation and jumps between distant frame regions while Docker resource usage was recorded.

Across the original and optimized architectures, the perceived annotation experience was similar. Normal frame navigation was sufficiently smooth for practical annotation work, while very aggressive scrubbing could produce brief stalls under both configurations. No clear user-facing degradation attributable to direct-disk frame serving was observed.

The controlled comparison therefore supports the primary goal of the optimization: **reducing CVAT infrastructure overhead while maintaining comparable annotator-facing responsiveness**.

4.1.3 SupportingStructure subsystem architecture

The SupportingStructure component is a human-in-the-loop computer-vision and annotation subsystem within the broader gastric-ultrasound pipeline. It is not simply a wrapper around a pretrained inference endpoint. Candidate generation is combined with deterministic image processing, geometric and topological validation, human anatomical review, controlled CVAT persistence, cryptographic authorization, duplicate protection, and independent readback verification.

A central design principle is the separation of generation from persistence. The system may generate candidate geometry during a dry run, but that geometry cannot be silently regenerated when annotations are later written to CVAT. Instead, the reviewer-approved result is frozen into a report and bound to an authorization using SHA-256. Only that exact approved geometry can proceed through the CREATE-only persistence path.

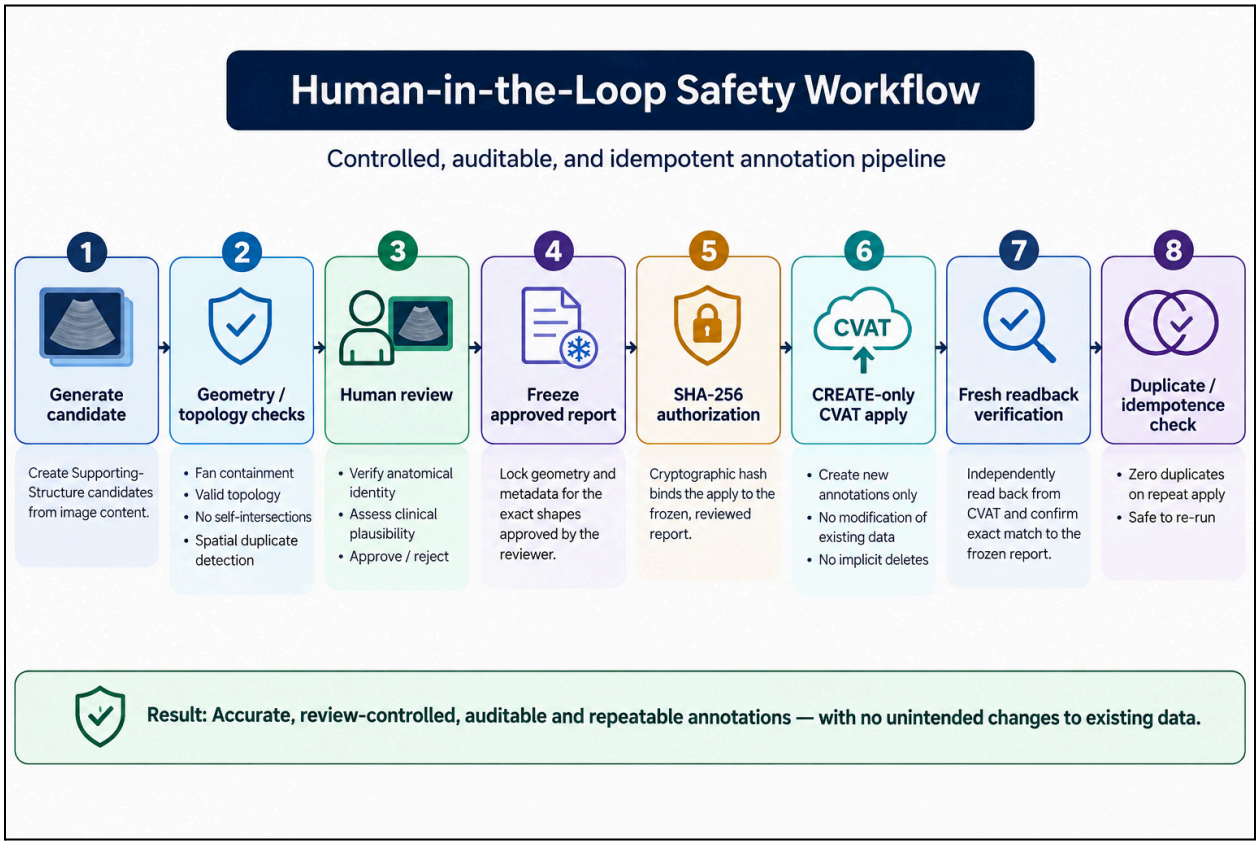


Figure 4. Human-in-the-Loop SupportingStructure Safety Workflow. Candidate generation is separated from persistence through geometry/topology validation, human anatomical review, frozen-report authorization, CREATE-only CVAT persistence, independent readback, and duplicate/idempotence verification.

4.1.4 Frame-selection Architecture

The frame-selection subsystem reduces the number of ultrasound frames passed to downstream annotation by prioritizing frames that are both visually usable and physiologically appropriate for gastric assessment. Each candidate frame is first evaluated using image-quality measures. When a usable temporal antral-area signal is available, the system applies contraction-aware selection to identify relaxation peaks corresponding to frames in which the antrum is near its maximum cross-sectional area between peristaltic contractions. The selected frame indices and associated scores are stored in a lightweight selection manifest for use by downstream preprocessing and annotation stages.

The architecture also includes an explicit fallback path for videos in which a reliable peristaltic signal cannot be identified. Rather than deriving relaxation peaks from an unreliable signal, the pipeline falls back to quality-based frame selection. This separation allows the same downstream annotation workflow to operate on both contraction-aware and quality-only selections while preserving the reason each frame was selected.

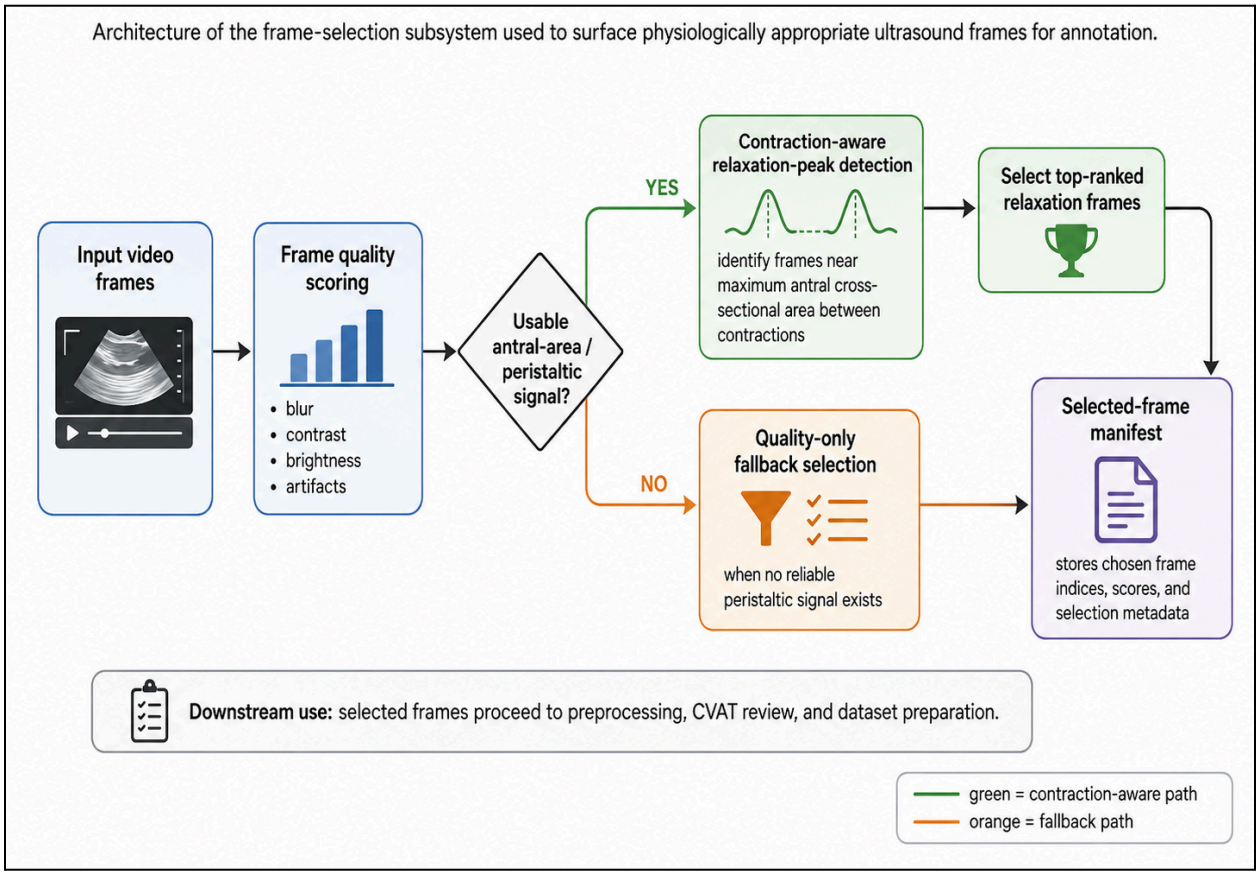


Figure 5: **Frame-selection architecture for gastric ultrasound annotation.** Video frames are first scored for image quality and assessed for a usable antral-area/peristaltic signal. When a reliable signal is available, contraction-aware relaxation-peak detection identifies and ranks physiologically appropriate frames; otherwise, a quality-only fallback is used. The selected-frame manifest records the chosen frame indices, scores, and selection metadata for downstream preprocessing, CVAT review, and dataset preparation.

4.2: Train / validation / test methodology

4.2.1 Antrum Train / validation / test methodology

In total, we have trained 4 different models .

Model Name	Dataset	Candidate model	Epoch
gold_label_finetune	gold_label(unmask class set)	navid-20260509-last.pt	33(stop as patience 15 is gone)
gold_label_finetune1	gold_label(class masked)	yolo26x_e120-best.pt	60(patience is 50)
gold_label_augmented*	gold_label(class_masked)	yolo26x_e120-best.pt	32 (as patience 15 gone)
review_gold_label_finetune	review_gold_label	yolo26x-seg.pt	14(as patience 10 gone)

*The augmented has utilized image augmentation utilities in ultralytic library which we set param of degree variation to 5,translate 0,05,scale 0,05,hsv-h 0,015,hsv-s 0,4,hsv-v0,3 aid our relative small dataset in gold_label .

4.2.2 SupportingStructure development and validation methodology

The SupportingStructure subsystem did not train a new project-specific neural network from scratch, so assigning its development cases to a conventional training/validation/test split would be misleading. Instead, its methodology progressed through four stages:

1. Exploration and candidate development: Tasks 1, 2, and 5 were used to evaluate early SupportingStructure candidates and expose failure modes.
2. Technical live validation: Tasks 31 and 33 exercised the end-to-end CVAT persistence workflow, including dry run, human approval, SHA-256 authorization, CREATE-only apply, independent readback, and duplicate/idempotence checks.
3. Uncertainty handling: Task 32 was intentionally not counted as a successful liver validation because the source anatomy was insufficiently clear. This demonstrated that the workflow can withhold a result rather than forcing an anatomical interpretation.
4. Systematic validation: Task 34 evaluates the final wall, liver, and pancreas methodology across five consecutive frames and freezes the approved results into permanent validation manifests.

Because Task 34 uses consecutive frames sourced from Task 1, it is not presented as independent generalization evidence. Additional materially different source tasks are required for a meaningful held-out evaluation.

4.3: Baseline comparisons

4.3.1 Antrum Baseline comparisons

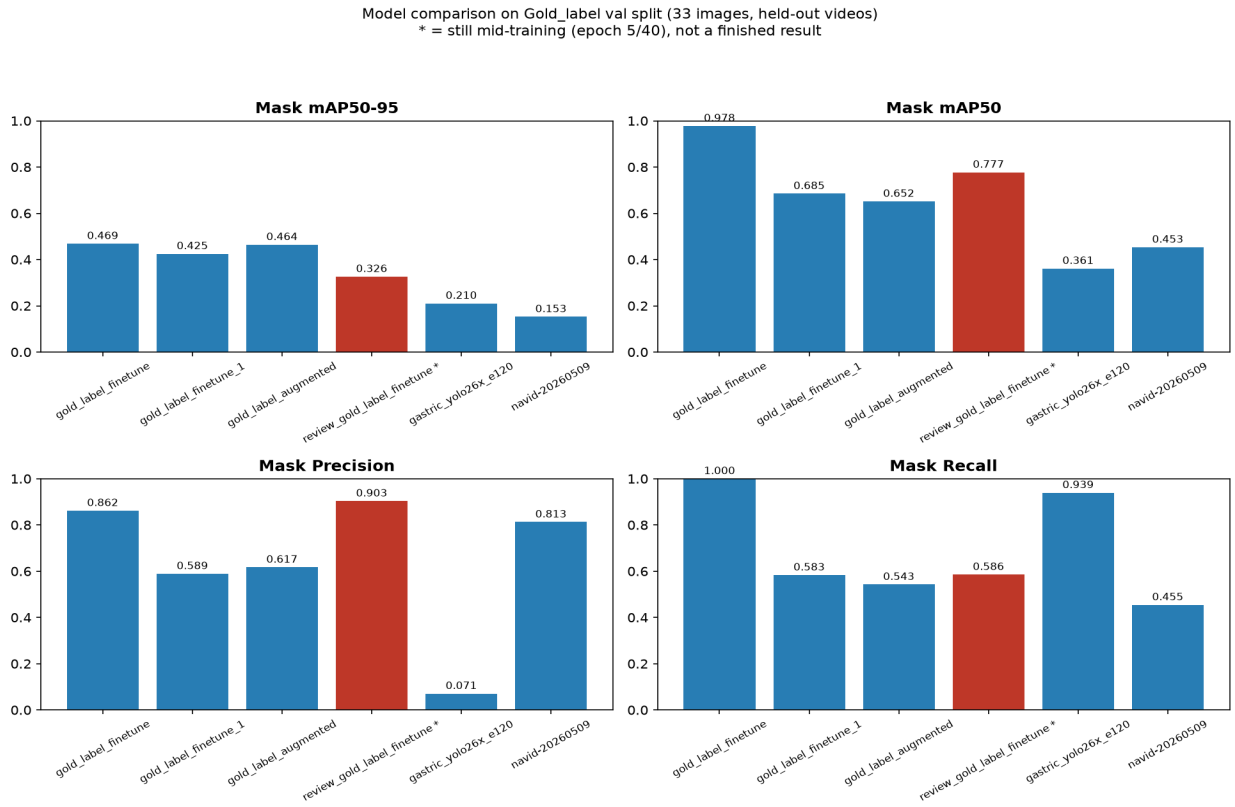


Figure 6 Metric comparison among models on the same val dataset

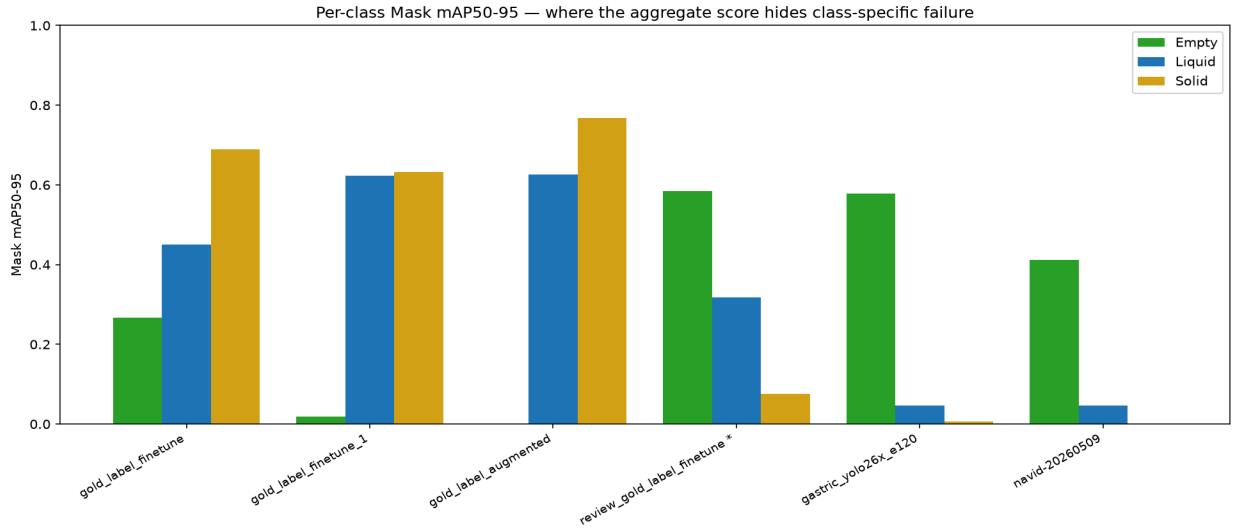


Figure 7: mAP50-95 metric comparison of performance of different model on different class

Navid-20260509 is the repo's current promoted production model, documented as "the default CVAT automatic annotation detector".Gastic-yolo26x is the earlier YOLO26x architecture reference, trained only on the 68-image seed set.Other 4 finetune model had undergoing training on top of these 2 model.

Mask mAP 50-95 serves as the primary pixel-level metric for evaluating how accurately the model segments the antrum boundary relative to human-annotated ground truth. The gold_review_augmented dataset variant achieved the highest performance on this metric. However, regarding **Mask Recall**—which measures the proportion of the true antrum region successfully segmented by the model—the `gold_label` fine-tuned model yielded the best results. While the `review_gold_label` fine-tuned variant also achieved high recall, its substantially lower Mask Precision indicates a tendency toward over-segmentation (i.e., generating excessively large or spurious masks, leading to a high false-positive rate).

As per class comparison, it showed the diverge trend, the model train using **Gold_label dataset** has significant lower accuracy versus the one has training dataset exposed to other data source, it could be due to that there is only 11 images of empty class frames in training dataset for **gold_label dataset,so there is not enough training for empty class for gold_label fine tune models.**

Generally, the fine tuned model has yielded better results than the candidate model or seed model as we are using the more accurate ground truth dataset for training.

4.3.2: SupportingStructure Baseline comparisons

SupportingStructure development included several methodological baselines that were retained long enough to explain why the final methods changed.

Abdominal wall: An earlier shallow-wall interpretation did not match the approved full anatomical extent. The final approach uses a deep, full-width wall constrained by a smoothed ultrasound-fan envelope.

Liver: Early SAM-derived and small anvil-shaped candidates could satisfy mechanical acceptance criteria while remaining anatomically incorrect. Strongest-edge and dynamic-programming traces could also lock onto visually stronger but anatomically incorrect interfaces. The final method therefore combines broad anatomical topology with image-derived interface tracing and human anatomical review.

Pancreas: An earlier fixed-width proposal imposed geometry rather than deriving the structure from frame content. The final method instead uses Otsu thresholding, morphology, connected-component selection, contour extraction, and simplification.

Acceptance baseline: Mechanical geometry checks alone were shown to be insufficient. Several early liver candidates passed automated checks but were rejected visually. The final system therefore uses automated geometry checks as safety gates while keeping human anatomical identity review authoritative.

4.4: Model optimization

4.4.1 Antrum Annotation

To systematically enhance segmentation performance, baseline evaluations were analyzed to diagnose and address key training bottlenecks. Diagnostic trajectories for gold_label_finetune revealed severe small-dataset overfitting, characterized by a steady decline in training loss (3.75 to 0.73) alongside a sharp increase in validation loss (3.0 to 4.8) after epoch 18. To mitigate this, an explicit --augment pipeline was integrated into train_yolo_seg.py . Augmentations reflecting realistic operational variability—such as small rotations (+-5 degrees), translation (5%), scale jitter (5\%), and brightness, contrast, and saturation adjustments—were enabled to simulate probe angle shifts and gain setting variations for creating more data samples in the relatively small dataset.

In addition to overfitting, evaluations highlighted severe training class imbalance within the original Gold_label split, where the Empty state accounted for only 9.1\% of training instances compared to 45.5% each for Liquid and Solid states. This disparity directly correlated with systemic mask collapse for the Empty class across baseline models. To resolve this, a rebalanced dataset variant (review_gold-label) was constructed by merging the 121 core Gold_label training frames with 3,179 additional frames from the larger reviewed pool. To preserve evaluation integrity, frames from the three held-out validation videos were strictly excluded to prevent data leakage, and duplicate frames defaulted to the primary verified Gold_label annotations. This strategy adjusted the training class distribution to 30.7% Empty, 38.2% Liquid, and 31.2% Solid while maintaining an untouched, leakage-free validation set. It should be noted as a methodological caveat that subsequent inspection raised questions regarding

the verification status of the supplementary source pool (navid-20260509_gus001_reviewed), presenting a dataset validity concern distinct from the underlying balancing logic..

4.3.2 SupportingStructure method refinement

Optimization of the SupportingStructure subsystem focused primarily on improving geometric reliability and anatomical plausibility rather than retraining model weights. Iterative improvements included fan-attachment validation, lower-edge topology checks, self-intersection rejection, connected-component handling, mask repair, contour-loop deduplication, spatial duplicate detection, temporal search priors across adjacent frames, and role-specific image-processing strategies.

Importantly, optimization was not performed by simply relaxing thresholds until a candidate was accepted. Candidates that were technically valid but anatomically implausible were retained as failure evidence and rejected by the human reviewer. This prevented automated metric optimization from overriding anatomical judgment.

4.5 Parallelization and frame selection

Motivation: The computational bottleneck in the annotation pipeline is SAM prelabeling, where a detection model and a segmentation foundation model run in sequence on every selected frame. A single-process run serializes this across all videos, leaving available CPU cores and memory idle while a single model instance works through one video group at a time. At the same time, the selection stage that precedes prelabeling must be decoupled from it: selection is a lightweight, CPU-only decision over quality scores and contraction-cycle timing, and coupling it to the GPU-bound prelabeling step would force the entire pipeline to reload models whenever selection criteria change. The architecture therefore separates the two concerns with a file-based contract—a JSON manifest that names exactly which frames enter prelabeling—and parallelizes only the expensive downstream stage.

Frame selection: Selection operates in three modes. In quality-only mode, the pipeline ranks every extracted frame by a composite image-quality score, with blur weighted at 50%, contrast at 30%, and brightness at 20%, and takes the top-K per video. In contraction-aware mode, it reads YOLO segmentation labels written by an earlier prelabeling pass, computes antral mask area per frame via the shoelace formula over polygon coordinates, smooths the resulting time series with a moving-average kernel, and detects local maxima using SciPy peak-finding. These peaks correspond to the antrum at maximum cross-section between peristaltic contractions—the measurement instant required by the Perlas protocol. Detected peaks are then ranked by quality score; when too few peaks survive quality gating, the remaining slots are filled by quality-ranked non-peak frames. The third mode is a per-video fallback: when a video yields no detectable contraction cycle due to a flat area signal, too few labelled frames, or a signal range below one percent of the image area, that video silently degrades to quality-only ranking while other videos in the same batch remain contraction-aware.

Best-frame selection: For the Perlas single-measurement use case, a further reduction selects exactly one frame per video. Priority is given to the relaxation peak with the largest antral area, with quality score used only as a tiebreaker. When no peaks are detected, the frame with the largest area anywhere in the video is chosen; when no area data exists at all, the highest-quality frame is returned. This single frame is the one a clinician would select manually for the cross-sectional area measurement that determines aspiration risk.

Parallelization: SAM prelabeling distributes work across multiple processes using Python's spawn multiprocessing context, which is safe for CUDA model loading and avoids the GIL. The unit of parallelism is the video group: each worker receives a disjoint round-robin partition of the video folders, loads its own YOLO and SAM model instances once on startup, and processes its assigned groups sequentially. Results are aggregated implicitly through the filesystem—each worker writes label files into the shared label tree, and because groups are disjoint, no inter-process communication or locking is required. A memory guard estimates available system RAM before spawning, reserving 1.5 GB for the system and budgeting 2 GB per worker, and caps the effective worker count to prevent out-of-memory failures on CPU runs. After all workers join, any non-zero exit code raises a RuntimeError, halting the pipeline rather than proceeding with partial labels.

Manifest contract: The selection manifest (selected_frames.json) is the sole interface between selection and prelabeling. It lists frame stems; the prelabeling step loads these stems into a set, filters each video group's images against them, and symlinks only the selected frames into a temporary directory before invoking the model. This design allows selection criteria to iterate without rerunning inference, permits the selection step to run without GPU dependencies, and ensures that the frames a human eventually annotates are exactly the frames the selection logic chose—no more, no fewer.

4.6: Pipeline correctness and automated testing

4.6.1 Antrum Annotation Pipeline Correctness

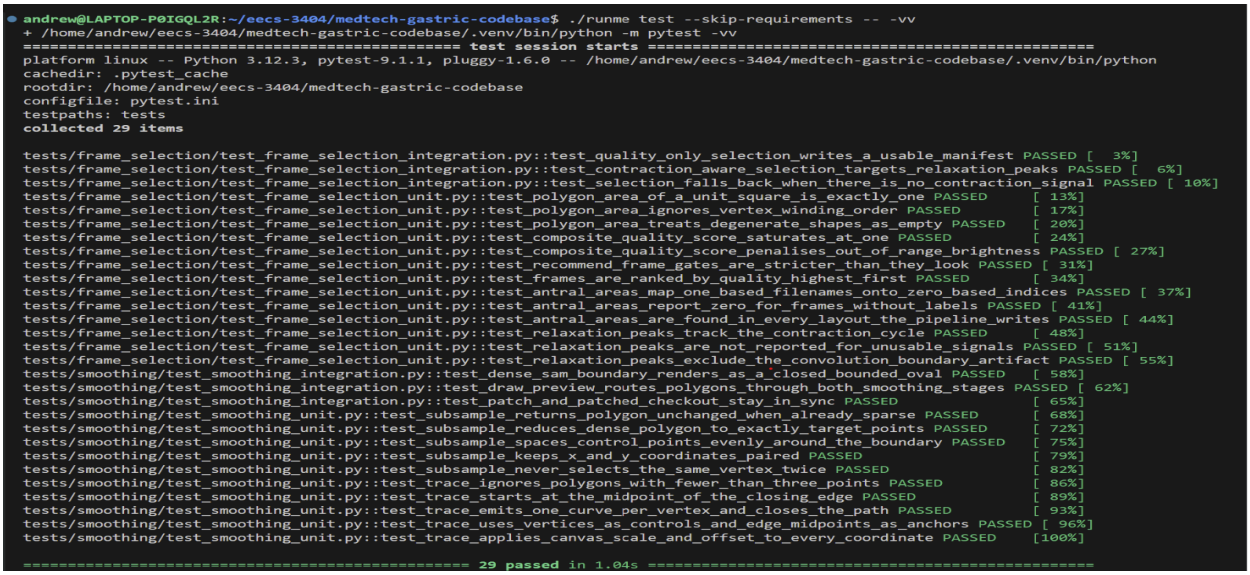
Motivation: The load-bearing failure mode in this project's data pipeline is a silent one. Frame selection determines which frames a human ever annotates, and therefore which frames enter the training set. A fault there does not raise an error, it produces a well-formed manifest of the wrong frames, which propagates into the training data unchallenged. The same applies to the annotation tooling, where a rendering regression produces a plausible but incorrect overlay that an annotator may accept. To constrain this, the project maintains a Pytest suite of 29 tests covering the two components where an undetected fault corrupts the dataset rather than halting the pipeline.

Frame selection: The unit tests pin the three layers the selection decision rests on: cross-sectional area measurement, composite image-quality scoring, and relaxation-peak detection. The peak-detection tests encode the clinical requirement directly per the Perlas protocol, antral area must be measured with the antrum at rest between contractions and at its widest asserting that detected peaks coincide with maximum antral area across a modelled contraction cycle, and that videos with no usable signal yield no peaks rather than inventing them from noise. Three integration tests drive the pipeline end to end, from quality report and prelabels on disk through to the written manifest, across all three operating modes: quality-only, contraction-aware, and the fallback for videos with no detectable peristalsis.

Annotation overlay: The overlay smoothing is implemented in TypeScript. Rather than reimplementing it for testing, the suite extracts the functions from the shipped source and runs them in an embedded JavaScript interpreter (QuickJS), so the tests exercise the code that actually runs in the annotation tool. A further test verifies that the committed patch and the patched checkout have not diverged.

Regression coverage: Two tests encode defects found during development that produced no error output: labels not being located in the directory layout the pipeline actually writes, which caused contraction-aware selection to degrade silently to quality-only; and zero-padding in the smoothing convolution, which manufactured artificial peaks at each video's boundaries.

Execution and scope: The suite runs against a test-only dependency set excluding the deep-learning stack, completing in roughly one second with no GPU, and writes exclusively to temporary directories, no test touches the dataset, model artifacts, or a live annotation server. It targets pipeline correctness, not model quality; model performance is assessed separately in Section 5.



Example run of the full Pytest suite, all 29 tests passing

4.6.2 SupportingStructure automated validation

SupportingStructure has a separate and substantially broader engineering test suite covering its candidate-generation, geometry, CVAT persistence, workflow-safety, duplicate-handling, CLI, and supporting data-utility paths. The final SupportingStructure PR stack was verified with **466/466 automated tests passing**.

These 466 tests should not be confused with the 29 tests described above. The 29-test suite specifically validates frame-selection and annotation-overlay correctness, whereas the 466-test result refers to the broader SupportingStructure development branch and its dependencies.

The SupportingStructure workflow also performs runtime safeguards that unit tests alone cannot provide: approved reports are SHA-256-bound, the authorization is rechecked immediately before CREATE, existing annotation fingerprints are preserved, newly written geometry is independently read back from CVAT, and repeat application is checked for idempotence.

5: Evaluation

5.1: Model evaluation and metrics

5.1.1 Antrum Detection Model Evaluation

These were the results of our best run:

Our model was trained for 32 epochs. Training losses decreased consistently across all components: train/box_loss reduced from 1.479 to 0.800 (46%), train/seg_loss from 2.917 to 0.608 (79%), train/cls_loss from 2.224 to 0.588 (74%), and train/sem_loss from 1.714 to 0.222 (87%). These reductions indicate that the model, including the auxiliary semantic head, converged on the training data.

The validation losses exhibit a different trend. The val/seg_loss reached its minimum of 2.701 at epoch 8 and subsequently increased to 7.281 by epoch 32. The val/box_loss reached its lowest value of 1.783 at epoch 9 before returning to 2.039 by the final epoch, while val/cls_loss and val/dfl_loss remained comparatively stable. The divergence between training and validation segmentation losses is consistent with overfitting in the mask branch. Peak validation performance occurred prior to the end of training: mAP50(B) reached 0.964 and mAP50-95(B) reached 0.527 at epoch 17, and mAP50-95(M) peaked at 0.486 at epoch 25. Both declined thereafter, indicating that the epoch 17 checkpoint, or epoch 25 if mask quality is prioritised, corresponds to the highest-performing model state.

A limitation of the evaluation is the size of the validation set. Recall values consistently aligned with discrete fractions such as 31, 32, and 3330, suggesting approximately 30 positive instances in total, which contributes to epoch-to-epoch metric variance. The learning rate schedule followed a three-epoch linear warmup to 1.42×10^{-3}, followed by cosine decay to 4.6×10^{-4}, consistent with the configured optimiser settings. In summary, the model achieved a peak detection performance of mAP50≈0.96, with mask branch overfitting observed beyond epoch 20.

epoch	metrics/precision(M)	metrics/recall(M)	metrics/mAP50(M)	metrics/mAP50-95(M)
1	0.68702	0.30303	0.35521	0.14565
2	0.02439	0.66667	0.47663	0.28878
3	0.59929	0.75758	0.69099	0.39481
4	0.90677	0.33333	0.49575	0.21876
5	0.87794	0.33333	0.67005	0.28419
26	0.58031	0.66667	0.66333	0.40831
27	0.54745	0.65113	0.66333	0.40428
28	0.59745	0.66667	0.64833	0.43219
29	0.60798	0.66667	0.64833	0.39165
30	0.50181	0.66667	0.62157	0.40478
31	0.47498	0.66667	0.66333	0.43345
32	0.56721	0.66667	0.63933	0.41431

5.1.2 SupportingStructure evaluation metrics

SupportingStructure is evaluated using metrics appropriate to annotation geometry and workflow correctness rather than classification accuracy alone. These include fan containment, polygon simplicity, anatomical adjacency, overlap measurements, connected-component behavior, CVAT readback fidelity, duplicate/idempotence behavior, and human anatomical approval.

Final Task 34 validation produced **20 persisted objects across five frames**: abdominal wall, liver, pancreas, and AntrumContext on each frame. All final wall, liver, and pancreas polygons were non-self-intersecting.

Abdominal-wall and pancreas fan-outside fractions were 0.0 across all five frames. Liver containment was also 0.0 on frames 1–3, with small accepted residuals on frame 0 (`0.0014748`) and frame 4 (`0.0001575`).

Liver–pancreas geometry demonstrated a useful validation exception. Frame 0 contained 2,569 pixels of overlap, equal to approximately **7.83% of the pancreas area**, demonstrating genuine disagreement between independently derived interface traces rather than a thin rasterization artifact. Later frames showed much smaller overlap.

Pancreas–AntrumContext overlap across frames 0–4 was **2.77%, 0.17%, 0.16%, 0%, and 0%**, respectively. Because AntrumContext is copied only after generation, this is a post-generation diagnostic and does not indicate an algorithmic dependency.

Live Task 34 annotations were independently read back and matched the five frozen approval manifests exactly. Task 1 remained unchanged at **518 shapes**, demonstrating that validation work did not modify the production source annotations.

5.2 Visualization

5.2.1 Antrum Detection Visualization

Figures 6 and 7 provide complementary visualizations of antrum-segmentation performance. Figure 6 compares the evaluated models on the same validation dataset using four mask-level metrics: mAP50–95, mAP50, precision, and recall. The comparison demonstrates that model performance cannot be characterized by a single metric. Fine-tuned models generally improved segmentation performance relative to the earlier seed/reference models, but the results also show trade-offs between boundary accuracy, precision, and recall. For example, a model achieving higher recall does not necessarily achieve the highest precision or mAP50–95, indicating that increased detection coverage can be accompanied by over-segmentation or additional false-positive mask regions.

Figure 7 disaggregates mask mAP50–95 by gastric-content class (Empty, Liquid, and Solid) and demonstrates why aggregate performance alone can conceal important class-specific failure modes. The original Gold_label training split contained substantially fewer Empty examples than Liquid or Solid examples, and the class-level results show correspondingly weaker Empty-class performance for several of the fine-tuned models. Conversely, models trained using the larger reviewed/rebalanced data show a different class-performance distribution, illustrating that improvements for one class do not necessarily transfer uniformly to the others. This visualization therefore complements the aggregate metrics in Figure 6 by exposing class imbalance and class-specific segmentation behaviour that would otherwise be hidden by a single overall mAP score.

Together, Figures 6 and 7 show that model selection should consider both aggregate segmentation quality and per-class behaviour. The results support the use of mask mAP50–95 as the primary overall segmentation metric while retaining precision, recall, and class-specific mAP50–95 as diagnostic measures for identifying over-segmentation, under-detection, and class imbalance.

5.2.2 SupportingStructure Visualization

Visualization is central to SupportingStructure evaluation because geometric validity cannot independently establish anatomical identity. Candidate overlays are therefore reviewed directly on the source ultrasound image before persistence.

The final report uses three complementary visualizations:

- the **Human-in-the-Loop Safety Workflow**, showing the controlled path from candidate generation to verified CVAT persistence;
- a **five-frame Task 34 validation view**, demonstrating wall, liver, pancreas, and AntrumContext across consecutive frames;
- selected **accepted and rejected candidate examples**, showing why automated geometric acceptance is not sufficient without human anatomical review.

These visualizations are intended to make both successful behavior and failure cases inspectable rather than reducing evaluation to a single aggregate score.

5.3: Diagnostics

5.3.1 Antrum Diagnostics

Antrum diagnostics include filename-label consistency checks, spatial duplicate detection via polygon IoU overlap, contour point-count and simplification diagnostics, YOLO label geometry validation (point count, coordinate bounds, single class per file), track keyframe and outside-boundary verification, subject-level split leakage checks, and fresh CVAT API readback for annotation-progress verification.

A particularly important diagnostic finding occurred during SAM/YOLO pre-labeling: candidates could produce geometrically valid, well-formed polygons while still carrying the wrong class label, since the detector had no access to the video-level ground truth encoded only in the filename. The filename-label guard caught and corrected 31 such mislabeled objects across the reviewed dataset. This established that geometric validity checks confirm a shape is well-formed but cannot independently confirm it is correctly classified. Human review and filename-derived correction therefore remain the final authority on label correctness.

5.3.2: SupportingStructure Diagnostics

SupportingStructure diagnostics include fan-containment and attachment checks, polygon simplicity and self-intersection detection, connected-component analysis, spatial duplicate detection, mask-repair diagnostics, contour-loop deduplication, topology checks, overlap measurements, SHA-256 report verification, fresh CVAT readback, and idempotence checks.

A particularly important diagnostic finding occurred during early liver development: candidates could satisfy mechanical acceptance criteria while still representing the wrong anatomical region. This established that geometry metrics provide necessary safety checks but cannot independently determine anatomical identity. Human visual review therefore remains the final authority for accepting a proposed SupportingStructure role.

6: Discussion

6.1: Limitations

6.1.1 Antrum Annotation Limitations

Annotator Expertise and Reliability

Lack of trained clinical professionals: Frames were annotated by student researchers rather than sonographers or radiologists trained in point-of-care gastric ultrasound. Accurately identifying the antral boundary, distinguishing rest from mid-contraction, and grading contents as Empty, Liquid, or Solid all require clinical pattern recognition that a written protocol alone cannot fully substitute for. This is the most significant threat to label quality, and several other limitations below stem from it.

No inter-annotator agreement or consensus review: Each frame was labeled by a single annotator, with no second reviewer and no adjudication step. Standard medical annotation practice typically reports an agreement statistic (e.g., Dice overlap, Cohen's kappa) across multiple raters to demonstrate reproducibility. Without that here, the consistency of these labels cannot be quantified.

Dataset Composition and Coverage

Single-subject dataset within the current annotation cycle: All annotated data to date comes from one subject (ID_01), even though ten additional subjects have already been collected and are awaiting annotation.

Sparse, non-exhaustive frame coverage: Roughly ten frames were annotated per 300-frame video, selected by an automated quality heuristic. Over 95% of frames were never labeled, and because selection favored clean, high-relaxation frames, the dataset underrepresents motion blur, awkward angles, and mid-transition states a deployed model would actually encounter.

Ground Truth and Validation Methodology

Linear interpolation does not model true peristaltic motion: Frames between manually-set track keyframes were filled by linear interpolation, not individually inspected. Since antral wall motion is nonlinear, interpolated frames may look plausible without being anatomically precise.

No validation against an independent clinical measurement: Labels follow the published Perlas frame-selection criteria. Whether this protocol-based gold standard agrees with an antral cross-sectional-area measurement taken independently by a qualified sonographer has not yet been confirmed.

6.1.2 SupportingStructure Limitations

The main SupportingStructure limitation is the current scope of systematic validation. The final combined abdominal-wall, liver, and pancreas methodology has been validated across five consecutive frames from one source clip. This provides useful regression and reproducibility evidence but is not sufficient to establish broad clinical or dataset-level generalization.

The system also validates geometric properties more strongly than anatomical truth. Fan containment, self-intersection, connectivity, overlap, and duplicate safety can be computed automatically, whereas deciding whether a candidate genuinely represents liver, pancreas, or another anatomical structure still requires human interpretation.

6.2: Uncertainty

6.2.1: Antrum Annotation Uncertainty

Label uncertainty: Every polygon and class assignment is stored as a single, hard ground truth, with no mechanism to flag boundary ambiguity, borderline contraction-state frames, or intermediate content grades (e.g., a small residual volume that is not clearly Empty or Liquid). Combined with the lack of inter-annotator agreement checking noted above, there is no way to distinguish a confidently-correct label from one an annotator was genuinely unsure about, both are treated identically as gold truth.

Detection-confidence uncertainty is discarded, not propagated: The SAM/YOLO pre-labeling step applies a fixed detection threshold (conf=0.25) to decide whether a frame receives a proposed polygon at all; frames below threshold receive nothing rather than a flagged low-confidence candidate. This is a hard accept/reject cutoff rather than a carried-forward uncertainty signal, so the resulting dataset has no record of which accepted detections were near the threshold (and therefore less reliable) versus confidently detected, and frames the model found genuinely ambiguous are simply absent rather than represented as hard examples.

Statistical uncertainty from sample size: With annotated data currently limited to a single subject, any performance figure computed from this dataset carries very wide, effectively unreported uncertainty, there is no basis yet for a meaningful confidence interval on generalization performance, and any single-subject metric should be treated as a point estimate with unknown variance rather than a stable measurement.

6.2.2: SupportingStructure Uncertainty

Ultrasound anatomy is inherently uncertain because image quality, acoustic shadowing, speckle, probe position, and tissue interfaces may make a structure ambiguous or partially invisible. The SupportingStructure workflow explicitly permits this uncertainty rather than forcing every role to be produced.

Task 32 is an example: the workflow itself completed correctly, but the available anatomy was not sufficiently clear to count the liver result as successful validation. Similarly, multiple early liver candidates were mechanically valid but visually rejected. These cases are recorded as uncertainty/failure evidence rather than being removed from the development history.

6.3: Drift

6.3.1 Antrum Drift

The gastric antrum is imaged in the sagittal epigastric plane, anterior to the aorta and inferior to the liver.

Point-of-care ultrasound is handheld: the operator's grip shifts, respiratory motion displaces the diaphragm, and the patient may move. Any of these factors can cause the scanning plane to translate away from the antrum, producing frames in which the target structure is partially sectioned, clipped by the field boundary, or absent entirely. Because the segmentation model was trained on well-centred antral views, it responds to a drifted plane not with a low-confidence detection, but with no detection at all—the frame receives no label file, and its antral area is recorded as zero. A sustained drift therefore appears in the contraction-aware time series as a run of zeros indistinguishable from a genuinely empty stomach or a model failure.

The pipeline does not detect drift explicitly. Instead, it responds to its consequences: when the area signal across a video is flat (range below one percent of the normalised image area), contains too few non-zero frames (fewer than twice the minimum peak count), or yields too few relaxation peaks, contraction-aware selection abandons peak-based ranking for that video and falls back to quality-only ordering. This fallback is per-video and silent—no warning is surfaced to the operator or annotator, and the selection manifest records only the method tag quality_only_fallback without distinguishing its cause.

The ambiguity has two downstream effects. First, a video affected by intermittent drift may still produce enough non-zero frames to pass the minimum-signal gate, in which case the surviving area measurements are treated as a valid contraction cycle. If the drift episodes happen to coincide with contractions, the resulting peaks reflect only the frames where the probe was on-target and relaxed. This may be correct by coincidence, or it may select from a biased temporal subset. Second, the best-frame-per-video mode, which picks the single frame with the largest antral area at a detected peak, cannot distinguish a frame that captures the antrum at its true maximum cross-section from one that captures it at an oblique angle with a spuriously large apparent area. Drift that rotates the scanning plane without fully losing the target can therefore inflate the measured cross-section in exactly the way the Perlas protocol measurement is designed to avoid.

The evaluation pipeline quantifies the symptom through detection rate—the fraction of frames per video on which the model produces any mask—but reports it as a summary statistic rather than flagging individual videos for review. A low detection rate is consistent with drift, an empty stomach in the Baseline condition, or a model that has not generalised to the patient's anatomy; the metric does not disambiguate these causes. No automated mechanism currently exists to distinguish them, flag a measurement as potentially drift-affected, or recommend that the operator re-acquire a video whose signal was interrupted.

6.3.2: SupportingStructure Drift

SupportingStructure performance may drift across patients, scanners, probe angles, patient positions, acquisition settings, and anatomical presentation. Changes in ultrasound brightness, contrast, field-of-view geometry, tissue boundaries, or image artifacts may affect thresholding, fan detection, and interface tracing.

The next validation phase should therefore use 2–3 materially different source tasks, not merely additional adjacent frames from the same clip. Per-source containment, overlap, acceptance, and failure metrics should be retained so that changes in behavior remain visible.

6.4: Leakage risks

6.4.1: Antrum Training Leakage Risks

There are three forms of leakage are relevant to Antrum:

Evaluation leakage: The eleven annotated frames in each Batch2 task are produced by linear interpolation between two manually-set keyframes spanning a single short temporal window (e.g., frames 277–287) of one video, so they are consecutive and highly correlated rather than independently sampled. They are suitable for a quick internal check of annotation quality within a video, but should not be split across train and validation, or reported as evidence of generalization, adjacent interpolated frames

are near-duplicates of one another, and a model evaluated on some of them after training on others would show inflated, misleading agreement.

Annotation-input leakage: Initial antrum polygons were generated by a YOLO/SAM model already trained on related gastric ultrasound imagery, before human review. Where an annotator accepted a proposed polygon with only minor adjustment rather than re-deriving the boundary independently, the resulting label partially reflects the prior model's output rather than an independent human judgment. Unlike a workflow that structurally excludes prior annotations from generation, this risk is not currently eliminated by the pipeline, the only safeguard is annotator review against the frame-selection guide, which is a process control rather than a structural one.

6.4.2: SupportingStructure Leakage risks

Two forms of leakage are relevant to SupportingStructure.

Evaluation leakage: Task 34 is derived from Task 1 and consists of consecutive, highly correlated frames. It is therefore suitable as a validation/regression reference but should not be reported as an independent held-out test set or evidence of generalization.

Annotation-input leakage: Existing gastric-content or antrum polygons could artificially simplify SupportingStructure generation if their geometry were used to construct the new structures. The final workflow avoids this dependency. Empty, Liquid, Solid, Baseline, and antrum annotations are not required generation inputs, and AntrumContext is introduced only after SupportingStructure generation for visual review.

6.5: Bias

6.5.1: Antrum Bias

The Antrum subsystem is subject to **reviewer bias** because human-reviewed CVAT annotations are treated as the operational ground truth for model training and evaluation. Although the annotation protocol was informed by consultation with gastric-ultrasound domain experts, the final frame annotations were produced primarily by student researchers and were not independently adjudicated by multiple trained clinical reviewers. Systematic errors in antral boundaries or gastric-content labels could therefore be learned by the model rather than identified as annotation errors.

This risk is increased by the absence of an inter-annotator agreement analysis or an independent clinical ground-truth comparison. Potential performance bias across **gastric-content classes, patient positions, subjects, and acquisition conditions** has also not yet been comprehensively quantified. Future validation should therefore include additional clinical review, multi-annotator agreement assessment, and broader subject and acquisition coverage before stronger claims of model generalization are made.

6.5.2: SupportingStructure Bias

Current SupportingStructure evidence can contain both **source bias** and **reviewer bias**. Systematic Task 34 validation covers one short source sequence, so image characteristics specific to that acquisition may be overrepresented. In addition, human anatomical approval reflects the interpretation of the reviewer and should not automatically be described as independent clinical ground truth.

Broader validation should therefore include materially different patients/scans and, where stronger ground-truth claims are required, agreement from an additional qualified reviewer or clinical expert.

7: Conclusion and Future Work

7.1: Antrum Annotation Conclusions and Future Work

The Antrum annotation subsystem progressed from an unlabeled video archive into a reusable, human-supervised gold-labeling workflow. It now combines quality-based frame selection, SAM/YOLO-assisted candidate generation, CVAT-based human review and correction, subject-level split infrastructure, and packaged export for downstream fine-tuning.

Fourteen videos for subject ID_01 were annotated to completion in this cycle, yielding 154 human-reviewed frames (11 per video) with reproducible manifests. The core workflow is built and working; the main remaining gap is multi-subject coverage rather than missing pipeline functionality.

The next step is to annotate 2–3 of the ten remaining subjects and treat that as the first genuine held-out generalization evaluation, since the existing subject-level split logic only activates once multiple subjects are present. Broader rollout and model fine-tuning should follow only after that validation.

7.2 SupportingStructure Conclusion and Future Work

The SupportingStructure subsystem progressed from exploratory candidate generation into a reusable, tested, human-supervised annotation workflow. It now combines image-derived candidate generation, geometry and topology validation, native CVAT persistence, human anatomical review, frozen-report authorization, independent readback, duplicate protection, and permanent regression evidence.

Task 34 provides the current systematic five-frame validation reference with 20 final persisted objects and reproducible approval manifests. The core workflow is built, tested, and working; the main remaining gap is broader generalization evidence rather than missing core workflow functionality.

The next step is to validate the final methodology on 2–3 materially different source tasks and treat those cases as the first meaningful held-out generalization evaluation. Larger-scale production rollout should follow only after that broader validation.

7.3 AI-Assisted Development Disclosure

Generative AI tools, including ChatGPT and Claude/Claude Code, were used to assist with code review, debugging, documentation drafting, and editing. AI-generated suggestions were reviewed, tested, and verified by team members before inclusion. The team remains responsible for the implementation, experimental interpretation, and final submitted content.

8: References

1. A. Perlas, C. Arzola, and P. Van de Putte, "Point-of-care gastric ultrasound and aspiration risk assessment: a narrative review," Can. J. Anesth., vol. 65, pp. 437–448, 2018.
2. L. Girón-Arango and A. Perlas, "Gastric ultrasound: a consistent tool for anesthesiologists amid the shifting sands of perioperative glucagon-like peptide-1 receptor agonist management," Can. J. Anesth., vol. 72, pp. 1608–1612, 2025.
3. T. Zou, H. He, J. Yang, Y. Wu, C. Lv, L. Zhao, and W. Yin, "Artificial intelligence real-time automated recognition of the gastric antrum cross-sectional area and motility rhythm via bedside ultrasound: a pilot study," Sci. Rep., vol. 15, p. 13883, 2025.
4. J. Liu, S. Li, M. Li, G. Li, N. Huang, B. Shu, J. Chen, T. Zhu, H. Huang, and G. Duan, "Development and validation of machine learning predictive models for gastric volume based on ultrasonography: a multicentre study," J. Clin. Anesth., vol. 107, p. 112010, 2025.
5. Y. Jin, M. Ma, Y. Yan, Y. Guo, Y. Feng, C. Chen, Y. Zhong, K. Huang, H. Xia, L. Yan, Y. Si, and J. Zou, "A convenient machine learning model to predict full stomach and evaluate the safety and comfort improvements of preoperative oral carbohydrate in patients undergoing elective painless gastrointestinal endoscopy," Ann. Med., vol. 55, no. 2, p. 2292778, 2023.
6. CVAT.ai Corporation, “Computer Vision Annotation Tool (CVAT),” computer software. doi: 10.5281/zenodo.4009388.
7. G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” computer software, Ultralytics, version 8.0.0, 2023.
8. G. Jocher, J. Qiu, M. Liu, S. Lyu, F. C. Akyon, and M. E. Kalfaoglu, “Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models,” *arXiv preprint arXiv:2606.03748*, 2026. doi: 10.48550/arXiv.2606.03748.
9. A. Kirillov et al., “Segment Anything,” in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.
10. P. Virtanen et al., “SciPy 1.0: Fundamental algorithms for scientific computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. doi: 10.1038/s41592-019-0686-2.
11. N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. doi: 10.1109/TSMC.1979.4310076.
12. FFmpeg Developers, “FFmpeg Documentation,” *FFmpeg Project*, software documentation, accessed Aug. 13, 2026.
13. Nuclio Project, “Nuclio Documentation,” *Nuclio*, software documentation, accessed Aug. 13, 2026.

Appendix A: Individual Team Member Contributions

Name - Student ID	Tasks
Laily Ajellu – 212319356	- Selected and scoped the project and coordinated the initial literature review, project documentation, shared Google Drive structure, client requirements, and recurring internal/client meetings. - Set up the personal development environment and analyzed the inherited codebase and existing CVAT/ML architecture. - Coordinated receipt and organization of the client dataset and prepared clarification questions regarding hardware, user interface, and workflow requirements. - Conducted initial manual gastric-ultrasound annotations with Dr. Sena and Dr. Perlas, incorporated clinical feedback into annotation requirements, and created initial UI mockups in Figma. - Coordinated with the University of Toronto student collaborators, assigned proposed tasks, and created a Work Breakdown Structure (WBS) to organize project responsibilities and progress. - Reviewed team pull requests, including David's CVAT optimization work, and benchmarked the original and optimized CVAT architectures to evaluate resource usage and annotation responsiveness. - Investigated and repaired a CVAT task-level chunk-routing defect discovered during optimization validation, added automated regression tests, and completed final Original-vs-Optimized benchmarking and documentation. - Updated the project pipeline documentation (`pipeline.html`) to reflect implementation priorities, architecture decisions, and the annotation/training workflow. - Extended the annotation pipeline to support surrounding anatomical structures, including the **liver, abdominal region, and aorta**, for training and display within CVAT. - Assessed opportunities for further CVAT infrastructure optimization and determined that additional reductions would require removal of functionality needed by the annotation workflow.
David Badiei – 219399328	- Set up the GitHub repo, organization, and the storage of model weights and datasets for the team (done through GitHub releases) - Developed the frame selection feature, which is able to detect the top n best frames in the video, either based on basic features like brightness, contrast, etc., or based on the Perlas framework - Optimized parts of CVAT, to make it less resource intensive and slow when doing annotation inferencing - Reviewed PRs for the GitHub repo that people other than myself created - Helped with project management tasks, such as organizing meetings and keeping track of other people's tasks - Worked on the video in regards to editing, presenting my part + the demo
Tran Tran – 220168829	- Conducted a literature review on gastric ultrasound analysis and relevant machine learning approaches. - Manually annotated the gastric antrum and surrounding major organs to establish anatomical references for the dataset. - Performed frame-by-frame gastric antrum annotation in CVAT to generate gold-standard labels for model training. - Reviewed SAM/YOLO annotations in CVAT. - Developed a script to retrieve and organize annotated frames for downstream model development. - Coordinated and conducted a meeting with Dr. Perlas to clarify and validate organ annotation procedures. - Led the assembly, technical writing, editing, and final polishing of the project report.
Andrew Kolosowski – 219713213	- Scoped out the existing PostgreSQL database and put together suggestions for building features on top of the existing architecture and boosting efficiency (some of which got implemented at later stages). Consistently tested and reviewed pull requests more over the course of the project. - Built the project's pytest test suite from scratch for testing frame selection and smoothing features

	- Corrected readme documentation to be up to date after various changes - Created the outline and plan for the slideshow presentation
John Xie – 218748947	- Develop the smooth reviewing alongside CVAT - Mask the videos - Construct dataset for training and validation - Running antrum training pipeline - Review and test PR throughout the life cycle of the project