Thesis Dissertation

COMPUTER-AIDED DETECTION OF THE ENDOMETRIAL LESIONS IN HYSTERO-SCOPIC IMAGING USING DEEP LEARNING

Arestis Konstantinou

UNIVERSITY OF CYPRUS

Department of Computer Science

May 2026

UNIVERSITY OF CYPRUS

Faculty of Pure and Applied Sciences

Department of Computer Science

COMPUTER-AIDED DETECTION OF THE ENDOMETRIAL LESIONS IN HYSTEROSCOPIC IMAGING USING DEEP LEARNING

Arestis Konstantinou

Supervisor

Dr Constantinos Pattichis

The Individual Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

May 2026

Declaration

I declare that this dissertation and any accompanying software are my own original work, conducted in full compliance with the University of Cyprus Code of Conduct for Research, available at https://www.ucy.ac.cy/legislation/kodikas-deontologias-kai-politikes/, and with full accountability on my part for the accuracy, integrity, and validity of all content.

Acknowledgements

I would like to express my sincere and genuine gratitude to my supervisor, Dr. Constantinos Pattichis, for his continuous guidance, encouragement, and support throughout the preparation of this thesis. His scientific expertise, constructive feedback, and commitment to high academic standards were fundamental to the completion of this work.

I am particularly grateful for his patience, availability, and willingness to provide direction at every stage of the research process. His comments and suggestions consistently helped me improve both the technical depth and the clarity of this thesis.

I would also like to thank him for the broader educational influence he has had during my studies. Through his teaching and mentoring, I gained a stronger appreciation for disciplined research, critical thinking, and academic integrity.

I am also sincerely grateful to my co-supervisor, Professor Vasilios Tanos, MD, PhD, for his clinical guidance, valuable feedback, and for providing hysteroscopic video material that supported the broader direction of this research.

In addition, I would like to thank the Department of Computer Science at the University of Cyprus for providing the academic environment, infrastructure, and support that made this work possible.

Finally, I wish to thank my family for their understanding, patience, and constant support throughout this demanding period. Their encouragement gave me strength and motivation to complete this thesis.

ABSTRACT

Computer-aided detection in hysteroscopy has strong potential to improve consistency in endometrial lesion assessment, yet performance depends on robust localization under class imbalance, visual variability, and clinically challenging imaging conditions. A YOLO11-based detection framework was developed for multiclass lesion localization in hysteroscopic images using HS-CMU and HS-CMU-V2 under a controlled fold-based protocol. The pipeline covers data harmonization, annotation consistency checks, cross-validation training, test evaluation, and qualitative visual analysis.

A structured sensitivity analysis was performed to quantify the effect of key configuration factors, including input resolution, augmentation policy, negative-sample ratio, learning rate, loss weighting, and model capacity. Results showed strong configuration dependence. Mosaic-based augmentation and tuned optimization improved detection quality, whereas higher input resolution and larger model capacity did not improve generalization in the present data setting. The engineered final configuration achieved the best overall profile with test mAP@0.5 of 0.81, test mAP@0.5:0.95 of 0.46, precision of 0.80, and recall of 0.76. An epoch-matched comparison at 100 epochs showed clear improvement over RetinaNet-ResNet50 across all reported test metrics.

These findings provide a reproducible baseline and practical guidance for clinically oriented hysteroscopic decision-support development.

Keywords: hysteroscopy, endometrial lesion detection, deep learning, YOLO11, computer-aided diagnosis

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CAD	Computer-Aided Diagnosis
CADe	Computer-Aided Detection
CADx	Computer-Aided Diagnosis / Characterization
CNN	Convolutional Neural Network
ROI	Region of Interest
YOLO	You Only Look Once
SAM	Segment Anything Model
HS-CMU	Hysteroscopic Dataset by Chaoyang Hospital and Capital Medical University
EC	Endometrial Cancer
EP	Endometrial Polyp
EPH	Endometrial Polypoid Hyperplasia
EH	Endometrial Hyperplasia without Atypia
AHE	Atypical Hyperplasia of the Endometrium
IFB	Intrauterine Foreign Body
CP	Cervical Polyp
SM	Submucous Myoma
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
IoU	Intersection over Union
AP	Average Precision
mAP	mean Average Precision
PR	Precision-Recall
F1	Harmonic Mean of Precision and Recall
NMS	Non-Maximum Suppression
HPC	High-Performance Computing
GPU	Graphics Processing Unit
CPU	Central Processing Unit
RAM	Random Access Memory
FPS	Frames Per Second
SLURM	Simple Linux Utility for Resource Management
SPPF	Spatial Pyramid Pooling Fast

1 Introduction

1.1 Endoscopy and Electronic Detection Assistance Systems

Hysteroscopy is a core endoscopic technique for direct visualization of the uterine cavity and is widely used for diagnosing and managing intrauterine pathology, including endometrial polyps, submucous myomas, endometrial hyperplasia, and malignant lesions. Compared with indirect imaging modalities, hysteroscopy provides immediate visual access to lesion morphology and supports targeted biopsy and intervention. Nevertheless, visual interpretation remains operator-dependent, and diagnostic consistency can be affected by lesion heterogeneity, variable optical quality, and differences in clinical experience.

In parallel with broader advances in medical computer vision, computer-aided systems for endoscopy have evolved from simple image-processing pipelines to deep learning approaches that support classification, localization, and segmentation tasks [6]. Within hysteroscopy, recent studies have shown promising results for automated lesion analysis, including multiclass endometrial lesion classification [7], deep-learning-based support for endometrial cancer diagnosis [8], and real-time endometrial polyp detection from hysteroscopic video [9]. More recent multicenter work has further reinforced the potential of AI-assisted hysteroscopic assessment in clinically relevant conditions [10].

Despite this progress, clinically reliable deployment is still challenging. Important limitations reported across the literature include limited dataset scale, class imbalance, inter-center variability, and incomplete standardization of evaluation protocols [7,9,10]. These constraints motivate reproducible detector-focused pipelines that provide interpretable outputs (e.g., bounding boxes and confidence scores), robust validation, and practical integration with existing clinical workflows.

From a clinical perspective, this direction aligns with the broader need for consistent classification and management of abnormal uterine bleeding etiologies, where structural causes such as

polyps, adenomyosis, leiomyoma, and malignancy/hyperplasia are central considerations in gynecologic decision making [11]. Therefore, AI systems in hysteroscopy should be designed not as autonomous diagnostic replacements, but as decision-support tools that improve consistency and reduce missed focal findings during image and video review.

1.2 Aims and Objectives

The main aim of this dissertation is to develop and evaluate a practical deep-learning pipeline for lesion detection in hysteroscopic images and video frames, with emphasis on reproducibility, interpretability, and clinically meaningful evaluation. The broader purpose is to help bridge the gap between recent deep-learning advances and practical computer-aided support in hysteroscopic diagnostics through a reproducible and clinically interpretable detection workflow.

A further objective is to assess detector robustness under realistic endoscopic image-quality conditions, including blur, specular highlights, bubbles and fluid artifacts, illumination variability, and partial occlusions. In addition, the study considers translational relevance by examining how this workflow can support more consistent lesion localization and potentially contribute to earlier identification of benign and malignant endometrial pathology during routine clinical review.

The specific objectives are:

1. Dataset integration and preparation: to organize and harmonize hysteroscopic data from HS-CMU and HS-CMU-V2 under a unified detection workflow and fold-based experimental design [1, 12].
2. Annotation and label consistency: to ensure valid image-label pairing, standardized YOLO-format annotations, and reproducible sample organization across train, validation, and test subsets.
3. Model development: to train a YOLO11-based detector for multiclass lesion localization in hysteroscopic media and optimize performance under realistic computational constraints [2, 13].
4. Quantitative and qualitative evaluation: to evaluate detection quality using precision, recall, mAP@0.5, and mAP@0.5:0.95 under cross-validation, complemented by frame-level and video-level qualitative analysis.
5. Preliminary robustness observations: to qualitatively examine detector behavior under realistic hysteroscopic image-quality variability, including blur, specular highlights, bubbles and fluid artifacts, illumination variability, and partial occlusions.
6. Clinical relevance assessment: to examine strengths and limitations of the developed sys-

tem as an assistive decision-support tool for hysteroscopic interpretation, rather than an autonomous diagnostic replacement.

1.3 Research Questions

This dissertation is guided by the following research questions:

1. To what extent can a YOLO11-based detector achieve reliable multiclass lesion localization in hysteroscopic images and video-derived frames under a reproducible cross-validation protocol?
2. How do data composition factors, such as class distribution, background prevalence, and sample heterogeneity, influence precision-recall behavior and overall detection robustness?
3. Which model and training design choices most strongly influence the trade-off between localization performance, stability across folds, and practical computational feasibility?

1.4 Contribution

The main contributions of this dissertation are as follows. An end-to-end hysteroscopy detection framework was developed to cover data preparation, training, validation, and inference within a reproducible pipeline. In addition, a unified multi-dataset protocol was established for HS-CMU and HS-CMU-V2 integration, with collision-safe sample handling and fold-consistent evaluation. Furthermore, a systematic detector evaluation setup was implemented using standard object-detection metrics and cross-validation aggregation, providing transparent and comparable performance reporting. Finally, a practical deployment-oriented analysis of model behavior on both static images and videos was conducted, with emphasis on clinically interpretable outputs.

1.5 Structure of the Thesis

The remainder of this dissertation is organized as follows. Chapter 2 reviews related work in AI-assisted hysteroscopy and identifies technical and clinical needs for robust detection systems. Chapter 3 presents the methodology, including datasets, annotation protocol, YOLOv11 configuration, training strategy, and validation metrics. Chapter 4 reports experimental results, including quantitative metrics and qualitative inference analysis. Chapter 5 interprets the experimental findings, compares results with other methods, and discusses limitations and practical implications. Chapter 6 presents the final conclusions and outlines future research directions.

1.6 Summary

In summary, this dissertation addresses the practical problem of hysteroscopic lesion detection by developing a reproducible YOLOv11-based CAD pipeline grounded in real clinical data and standardized evaluation. The work focuses on localization performance, robustness across heterogeneous data conditions, and clinically interpretable outputs, aiming to support more consistent and efficient hysteroscopic assessment.

2 Background of Hysteroscopy Detection Systems

2.1 Literature Review of Related Bibliography on Detection Systems in Hysteroscopy

Hysteroscopy is a key technique for the diagnosis and management of endometrial lesions (e.g., endometrial polyps, endometrial hyperplasia, endometritis, intrauterine adhesions, intracavitary myomas/fibroids). In recent years, the application of artificial intelligence (AI) to hysteroscopic imaging has made impressive strides, with computer-aided systems designed to assist in the detection, localisation and classification of such intrauterine abnormalities, with the goal of improving diagnostic accuracy, standardising examination quality, and supporting clinical decision-making.

AI systems in hysteroscopy encompass a range of functionalities, from detection and segmentation to classification, with the objective of achieving a more precise diagnosis and treatment of diseases.

Classification is a supervised machine learning task in which a model is trained on labeled data to determine which predefined classes are present in a given image, video, or other type of data. In clinical applications, endometrial lesions in hysteroscopic images can be classified into categories such as normal endometrium, endometrial polyps, endometrial hyperplasia and endometritis. Each category is associated with a specific treatment. Classification results can therefore support clinicians in making more accurate and consistent diagnostic desicions. Furthermore, classification methods may also be employed to assess the quality of hysteroscopic imaging itself, for instance by categorising frames according to optical clarity and adequacy of visualisation of the endometrial cavity.

Object detection is a computer vision task that combines classification and localisation to deter-

mine which objects are present in an image or video frame and where they are located. In practice, bounding boxes are typically used to delineate and distinguish objects in images or video frames. In hysteroscopic imaging, object detection can be used to highlight suspicious findings such as endometrial polyps, intrauterine adhesions, intracavitary myomas and to identify key anatomical landmarks such as the tubal ostia and fundus. Such detections can help reduce the risk of missing focal lesions, standardise visual inspection, and provide region proposals for subsequent evaluation.

Image segmentation is the process of partitioning an image into regions at the pixel level, with borders and shapes, delineating potentially meaningful areas for further processing, such as measurement, classification, and object detection. The regions may not cover the entire image, but the goal of segmentation is to simplify and transform the representation of an image into something that is more meaningful and easier to analyse. Segmentation differs from classification and object detection in that it provides pixel-by-pixel delineation of structures or lesions rather than assigning labels to entire images or coarse bounding boxes. For example, in hysteroscopy, segmentation of endometrial lesions can be used to estimate lesion size, shape, and extent within the endometrial cavity, which are important factors for diagnosis, treatment planning and follow-up evaluation.

The literature review reveals that AI systems for hysteroscopy have the potential to increase the detection rates of focal endometrial lesions, improve the accuracy of endometrial pathology categorisation and provide valuable information on lesions and morphology through segmentation. Despite promising developments, the application of AI in hysteroscopy still faces significant challenges, including the limited availability of annotated datasets, variability in image acquisition and optical quality, data security, patient privacy, and seamless integration into routine clinical practice.

A key aspect of the review is the development and adoption of standards for evaluating and comparing AI systems, underscoring the need for consistent and standardized performance assessment. Successful translation of these technologies into clinical practice requires close collaboration among clinicians, software engineers, and regulatory authorities. Moreover, building large and diverse real-world clinical datasets and establishing benchmarking frameworks that assess model performance across varied clinical conditions are essential steps toward ensuring generalizability and robustness.

In addition, the bibliograph emphasizes the ethical and legal challenges associated with deploying AI in medical practice. Common themes include patient privacy and data security, transparent and traceable model decisions, and bias mitigation to support reliable diagnosis. The literature frames these as prerequisites for responsible clinical adoption, alongside robust validation and standardized evaluation.

Overall, the literature review shows that AI-assisted hysteroscopy can support the detection, localization, and characterization of intrauterine pathology, with the potential to improve diagnostic consistency and reduce missed focal lesions. However, successful integration into routine clinical practice remains limited by small and heterogeneous datasets, variability in image acquisition and optical quality, class imbalance and the scarcity of external and prospective validation. These limitations highlight the need for robust data collection, standardized evaluation protocols, and clinically aligned performance targets. The following section broadens the discussion to related detection systems and extracts the key requirements and needs.

2.2 References to Existing Detection Systems and Identified Needs

This section provides an overview of current artificial intelligence (AI) computer-aided systems developed for hysteroscopy, focusing on methods that support the detection, localization, and characterization of intrauterine pathology during procedures or through offline analysis, with the broader aim of enabling clinically deployable decision-support tools. The reviewed approaches include task-specific solutions for endometrial polyp detection, discrimination of atypical endometrial hyperplasia and endometrial cancer, and segmentation of relevant anatomical structures, reflecting steady progress toward practical clinical use.

2.2.1 Existing Detection Systems

Recent studies have demonstrated diverse AI-based approaches for hysteroscopic image and video analysis, spanning classification, detection, and segmentation tasks. In multiclass lesion classification, Shengjing Hospital of China Medical University (2021) reported a CNN-based system for automated categorization of common endometrial lesions, aiming to provide objective diagnostic support during hysteroscopic assessment [7]. In cancer-focused analysis, the University of Tokyo proposed a deep learning framework for automatic detection of endometrial-cancer-affected regions in hysteroscopic images, with emphasis on image-based decision support [8].

For real-time lesion detection, Maternal and Child Hospital (Hubei) and Tongji Hospital (2023) introduced an improved YOLOX-based pipeline for endometrial polyp detection, incorporating group normalization for large-image handling and adjacent-frame association post-processing to stabilize predictions across video frames [9]. Similarly, the University of Porto and multicenter collaborators (2025) developed and validated a deep learning system for automatic detection and classification of endometrial polyps in hysteroscopy videos, using bounding-box localization to improve clinical interpretability and support potential real-time use [10].

Segmentation-oriented contributions have also been reported. Burai et al. presented a deep learning approach for automatic uterine-wall segmentation in hysteroscopy frames, providing a consistent anatomical region of interest for downstream computer-aided analysis [14]. In related clinical-assessment applications, Beijing Obstetrics and Gynecology Hospital (Capital Medical University) reported a deep learning system for evaluating endometrial injury and intrauterine adhesions, including a visualization panel intended to support outcome-related interpretation [15]. Finally, Török and Harangi (2018) described a deep learning-based support method for hysteroscopic fibroid resection, designed to delineate the boundary between submucous myoma and normal myometrium and thereby improve intraoperative visual guidance [16].

These AI based systems demonstrate the potential of advanced machine learning models to enhance hysteroscopy by improving the accuracy and efficiency of diagnostics, thereby supporting better patient outcomes and more streamlined clinical workflows.

2.2.2 Identified Needs for Advancing Hysteroscopy Detection

Despite hysteroscopy being the gold standard for evaluating and treating intrauterine conditions, several technical and clinical requirements must be addressed to enable reliable and clinically deployable AI assisted hysteroscopy systems.

Improving the detection of focal lesions remains a central requirement in AI-assisted hysteroscopy. A major challenge is that focal intrauterine lesions may be missed during routine assessment, particularly when they are small or visually subtle. Accordingly, there is a clear need to develop CADe-style tools that provide reliable detection support, either in real-time or offline settings, in order to reduce missed lesions [9].

A second requirement concerns consistency in detection. Detection and interpretation can vary across clinicians due to differences in experience and subjective assessment. Therefore, decision-support systems should provide consistent outputs, such as stable bounding boxes and confidence scores, across operators and clinical settings [10].

Another important need is the objective standardization of lesion characterization. Visual differentiation among intrauterine pathologies can lead to inter-observer variability and inconsistent clinical decisions. For this reason, CADx-style models that assist classification and characterization are needed to support more objective and reproducible assessment.

Higher-quality training data and annotations are also essential. Many studies rely on limited, heterogeneous datasets, while AI models require accurately labelled, high-quality images that are difficult to compile. As a result, comprehensive datasets and standardized annotation protocols are needed, whether based on clinical reports or expert subjective review, to support effective model training.

Robustness under real-world imaging conditions is another key requirement. Hysteroscopic videos often contain artifacts (e.g., bubbles) and variable visibility, both of which degrade model and human performance. This highlights the need for models and pipelines that explicitly handle artifact and quality variation, for example through stabilization across frames and artifact-aware preprocessing, to improve reliability in routine conditions.

Finally, seamless integration into clinical practice remains critical. Introducing AI tools into established medical workflows can be disruptive and may face resistance. Consequently, AI systems should be designed to integrate smoothly with existing hysteroscopy equipment and protocols, provide interpretable outputs, and support workflow-compatible deployment while adhering to ethical and privacy requirements.

By addressing these challenges and meeting these needs, AI applications for detecting intrauterine pathologies can substantially improve hysteroscopic assessment, supporting earlier and more accurate identification of causes of abnormal uterine bleeding such as polyps, submucosal fibroids and endometrial hyperplasia and thus enabling more appropriate clinical management.

2.3 Summary of Literature and Research Gap

The literature demonstrates growing interest in computer-aided support for intrauterine pathology evaluation, covering multiclass classification of endometrial lesions [7], localization of endometrial cancer affected regions, [8], real time detection of endometrial polyps from hysteroscopic video, and segmentation based support for anatomical understanding and guidance [14, 16].

Despite these advances, key gaps remain. Many approaches are still restricted by limited and heterogeneous datasets, differences in labelling and evaluation protocols, and insufficient testing across varied clinical conditions, which complicates fair comparison and generalization. In addition, clinically deployable systems require explainable outputs and workflow compatible design, together with evaluation on representative real world cases. [10, 15]

Motivated by these limitations, this work explores the use of modern detection and segmentation paradigms specifically a YOLO based detector (YOLOv11) for lesion localization [17] and a base segmentation model such as the Segment Anything Model (SAM) for mask generation and refinement [18]. The main question is whether such components can be adapted to hysteroscopic imaging to provide robust, interpretable outputs under realistic acquisition variability while maintaining performance that aligns with clinical needs.

3 Methodology

3.1 Report and Use of YOLO11 in Object Detection

This sub-chapter presents the application of YOLO11 (You Only Look Once, version 11), a deep learning architecture for lesion detection in hysteroscopic data. YOLO11 is a one-stage object detector that jointly performs localization and classification in a single forward pass, producing bounding boxes, class predictions, and confidence scores. The objective is to leverage YOLO11's capabilities to improve detection of abnormal regions of interest in both still images and video frames, thereby providing spatially explicit and clinically interpretable outputs, compared with traditional methods and earlier AI models.

3.1.1 Introduction to YOLO11

YOLO11 belongs to the YOLO family of real-time object detectors, which have undergone substantial architectural refinement since their original introduction. The version employed in this work incorporates an updated backbone and neck design that improves feature extraction

across multiple spatial scales, making it particularly well-suited for detecting lesions that vary considerably in size and appearance, as is common in hysteroscopic examinations [17].

Unlike multi-stage detectors that decouple region proposal from classification, the single-pass design of YOLO11 reduces inference latency without sacrificing localization accuracy, a meaningful consideration when processing large collections of video-derived frames or when targeting near real-time clinical feedback [9].

A further property relevant to this application is the model's behaviour under class imbalance. Hysteroscopic datasets are characterised by large disparities between common findings such as endometrial polyps and rare but clinically critical lesions such as atypical hyperplasia of the endometrium. The loss formulation of YOLO11, combined with appropriate augmentation strategies, allows training to be guided toward minority classes without discarding the broader detection capacity of the model [10].

In this dissertation, YOLO11 is used in detection mode to produce bounding boxes, class labels, and confidence scores for hysteroscopic lesion categories. The resulting predictions are evaluated using precision, recall, mAP@0.5, and mAP@0.5:0.95 under cross-validation, and are further used for frame-level and video-level qualitative assessment.

3.1.2 Usage of YOLO11 Object Detection

The process of custom object detection using YOLO11 involves a series of critical steps that were followed to configure and train the model effectively. Reliable detection performance requires careful preparation, beginning with high-quality data and consistent annotations. In addition, organizing samples into the required folder structure and generating fold-specific `data.yaml` files are essential components of the pipeline. The Ultralytics framework was used to streamline model training, validation, and testing.

Step 1: Data Collection
At the beginning of the workflow, two datasets were used: HS-CMU (primary dataset) [1] and HS-CMU-V2 (additional dataset). HS-CMU corresponds to the initial public dataset release, while HS-CMU-V2 was introduced later in this work as an extended dataset root. The second dataset was integrated to increase sample diversity and include mostly challenging hard-negative content, improving robustness under more varied visual conditions.

Step 2: Data Annotation and Label Consistency
All samples were prepared in YOLO detection format, where each image is associated with a text file containing class ID and normalized bounding-box coordinates. Before training, annotation consistency was checked to ensure valid image–label correspondence and prevent malformed labels from affecting optimization.

Step 3: Folder Setup and Sample Organization
To maintain an organized training workflow, fold-specific directory structures were created for `train`, `val`, and `test` subsets, each containing `images/` and `labels/`. In the implementation, samples were copied into `run_XX/fold_YY/{train,val,test}/{images,labels}`. When combining multiple dataset roots, unique sample identifiers of the form `R{root_id}_{stem}` were assigned to avoid filename collisions.

Step 4: YAML Configuration
For each fold, a `data.yaml` file was generated automatically. This file defined dataset paths, number of classes, and class names, and served as the configuration interface between the prepared data and the Ultralytics training framework [4].

Step 5: YOLO11 Training
Using the Ultralytics library, YOLO11 was loaded in detection mode and trained with predefined hyperparameters (image size, batch size, learning rate, optimizer, and augmentation settings). Cross-validation was employed to train and evaluate the model across folds under a controlled experimental setup. The complete set of baseline hyperparameters is reported in Section 3.3.2.

Step 6: Performance Evaluation
After training, each fold's best checkpoint (`best.pt`) was evaluated on validation and test subsets. Performance was measured with precision, recall, mAP@0.5, and mAP@0.5:0.95, and aggregate statistics (mean and standard deviation across folds) were used for model comparison and selection.

Step 7: Image and Video Inference Analysis
Finally, the selected model was applied to both static images and unseen videos. For video analysis, detections were produced frame by frame with bounding-box overlays and confidence values, enabling qualitative inspection of model behavior in realistic hysteroscopic sequences.

Through these steps, a complete YOLO11-based detection pipeline was established, covering data preparation and loading, training, quantitative evaluation, and qualitative clinical-style assessment.

3.1.3 YOLO11 Object Detection in Images

YOLO11 is widely used for object detection in static images due to its one-stage design, where localization and class prediction are performed in a single forward pass. Built on advanced deep learning methodologies, it combines convolutional feature extraction, multi-scale feature aggregation, and attention-enhanced modules (e.g., C3k2/C2PSA/SPPF) to produce a bounding box, class label, and confidence score per detected object, enabling efficient and interpretable predictions in practical computer vision workflows [3, 4].

This functionality is highly beneficial across a wide range of sectors. In surveillance, YOLO11 can assist in identifying potential security threats from camera streams. Within the medical field, it supports the detection of abnormalities in diagnostic images, contributing to earlier clinical assessment and treatment planning. In autonomous systems, including drones and UAV platforms, YOLO11 enables real-time scene understanding for obstacle awareness, target detection, and navigation support. It is also widely applicable in industrial inspection, traffic monitoring, and retail analytics, where rapid and reliable object-level interpretation is essential [3].

By efficiently processing and analyzing static images, YOLO11 provides structured visual outputs that improve situational awareness, support data-driven decisions, and enhance operational effectiveness across these domains.

YOLO11 Image Classification: One of the core features of YOLO11 in image detection is image classification. By analyzing the content of an image, YOLO11 can successfully assign a specific class or label to each detected object. This process enables thorough categorization of the objects within an image, supporting downstream analysis and informed decision-making based on the detected classes. For example, in computer-aided diagnosis (CAD) systems, YOLO11 can classify detected objects into clinically meaningful lesion categories, such as polyps, benign masses, precancerous abnormalities, and malignant findings. This structured output provides valuable decision-support information for clinicians during image review and assessment.

Object Localization and Recognition: In addition, beyond class prediction, YOLO11 performs object localization by accurately estimating bounding-box coordinates for each detected object in the image. These bounding boxes provide explicit spatial information about object position and extent, which is necessary for downstream interpretation, review and decision support. In practical terms, localization transforms detection from a purely categorical output into a visually grounded representation of findings. Object recognition is achieved conjointly with localization by assigning a class label and confidence score to each predicted box. This integrated prediction mechanism enables precise identification of multiple objects within a single image while preserving computational efficiency through one-stage inference. In comparison with traditional pipelines that often separate feature extraction, localization, and classification into multiple stages, YOLO-based detectors provide a more direct and operationally efficient workflow for real-time and high-throughput applications [3, 4, 17]. These capabilities are widely applicable across domains such as surveillance, autonomous systems and healthcare, where both what is detected (recognition) and where it appears(localization) are equally important for reliable analysis and action.

Table 3.1: Conceptual comparison between traditional detection pipelines and YOLO-style detection.

Criterion	Traditional / Multi-stage Methods	YOLO-style One-stage (YOLO11)
Pipeline structure	Detection is typically decomposed into proposal and classification stages or separate modules.	Localization and class prediction are produced in a unified, end-to-end pass.
Inference behavior	Higher pipeline complexity and typically higher latency due to staged processing.	Designed for real-time operation with a favorable speed-accuracy trade-off.
Output formulation	Region proposals and class decisions are often refined across multiple steps.	Direct output of bounding boxes, class labels, and confidence scores from one detector head.
Deployment practicality	More engineering overhead for integration and optimization in production.	Simpler operational workflow for high-throughput and real-time applications.

The comparison in Table 3.1 highlights the advantages of YOLO over traditional methods, showcasing its efficiency and effectiveness in its crucial tasks and deployment.[3, 6, 19].

3.1.4 YOLO11 Object Detection in Videos

Beyond static-image analysis, YOLO11 is not limited to still images. It can also be deployed on continuous video streams to perform real-time detection. In a standard deployment setting, the model processes the video as a sequence of frames and performs detection on each frame, producing bounding boxes, class labels, and confidence scores. This frame-level formulation supports consistent localization across dynamic scenes while ensuring efficient inference behavior.

In practical workflows, standard container formats such as MP4, AVI and MPEG are natively supported, lowering the barrier to deployment in existing production pipelines. In this implementation, the video is processed frame by frame, and the predicted bounding boxes, class labels, and confidence scores are overlaid on the reconstructed output stream, allowing visual verification of model predictions.

From a methodological perspective, the detection of video objects has also been extended with temporal aggregation strategies that exploit information from neighboring frames. Such methods improve robustness under motion blur, occlusion, and rapid scene changes by combining short-term and long-term context [20, 21]. YOLO11 provides a practical one-stage baseline for

this setting, balancing detection quality with computational efficiency in video-oriented applications [3].

Operational Video Inference Pipeline:

1. Input: A video stream (e.g., MP4/AVI/MPEG) is loaded and decoded into individual frames.
2. Per-frame inference: Each frame is passed through YOLO11 to produce candidate detections (bounding boxes, class labels, and confidence scores).
3. Post-processing: Detections are filtered using a confidence threshold, and temporal stabilization may be applied across consecutive frames to reduce visual jitter.
4. Output: Annotated frames are re-encoded into an output video for straightforward review of detection consistency over time.

Overall, video object detection generalizes image-based detection into the temporal dimension while retaining the same fundamental prediction outputs (bounding boxes, class labels, and confidence scores). In practice, inference operates at the individual frame level. In this work, a simple moving-average buffer over neighboring frames is applied to smooth confidence scores and stabilize the final predictions. More sophisticated temporal aggregation strategies, such as feature aggregation and memory based methods, have been shown to further improve robustness under motion blur, occlusion, and abrupt appearance changes [20, 21], though these lie beyond the scope of the present implementation.

3.2 Datasets and Annotation Protocol

3.2.1 HS-CMU Dataset

The principal dataset employed in this research is HS-CMU, originally presented by Han et al. for diagnostic analysis of benign and malignant conditions in hysteroscopy [1]. As the officially published release, HS-CMU constitutes the core data source of this implementation and defines the baseline lesion distribution used for model development.

For reproducibility purposes, key dataset properties reported in the original publication (e.g., number of patients/cases, total images, and class composition) are adopted as the reference profile of the dataset [1]. In this work, HS-CMU samples were integrated into a YOLO detection workflow by maintaining one-to-one image label correspondence and by structuring data into fold-specific train/validation/test subsets under a controlled cross-validation framework. Annotations were not created in this implementation. Author-provided labels from the dataset release were used directly.

According to the original dataset report, HS-CMU was collected between August 2023 and January 2024 at Beijing Chaoyang Hospital (Capital Medical University), and includes 175 videos from 175 patients. Video acquisition was performed with a Karl Storz TC200 endoscopic camera system at a resolution of 1920×1080, with recordings stored in MP4 format. The released HS-CMU package already provides the extracted annotated image frames and corresponding YOLO label files used in this work. The published release contains 3,385 labeled images across eight lesion classes (SM, EC, EP, EPH, EH, IFB, CP, AHE) with YOLO-format labels (`id cx cy w h`). As reported in the source statistics, the dataset is imbalanced, with EP and EPH among the most represented classes and AHE the rarest class [1]. The imbalance-mitigation strategy adopted in training is described in Section 3.3.2.

Table 3.2: Class-wise statistics of the HS-CMU dataset (adapted from [1]).

Label (ID)	SM (0)	EC (1)	EP (2)	EPH (3)	EH (4)	IFB (5)	CP (6)	AHE (7)
Images num	287	268	1059	574	280	404	537	83
Bounding box num	325	352	1330	683	383	466	550	96

* SM, submucous myoma; EC, endometrial cancer; EP, endometrial polyp; EPH, endometrial polypoid hyperplasia; EH, endometrial hyperplasia without atypia; IFB, intrauterine foreign body; CP, cervical polyp; AHE, atypical hyperplasia of the endometrium.

As summarized in Table 3.2, this imbalance profile motivated train-only negative downsampling and class-loss reweighting in the training stage Section 3.3.2.

3.2.2 HS-CMU-V2 Dataset

To broaden the training data beyond HS-CMU, a second dataset root, referred to here as HS-CMU-V2 was incorporated as a complementary resource. Rather than modifying the original HS-CMU, this second version was provided as an official online update from the same data source, rather than as a separately peer-reviewed publication, and therefore, it is cited as a dataset release resource[12].

The original HS-CMU release in April 2024 contained 3,385 annotated images drawn from 175 videos and 175 patients, while a later update in June 2025 contributed 2,405 images from 106 additional videos and 106 patients, including partially normal frames. Among the 2,405 update images, 1,184 label files are empty and 1,221 are non-empty, which appear as empty-label samples under the YOLO convention when no lesion annotation is present. The cumulative release reports 5,790 images from 281 videos and 281 patients [12].

Since both folders share the same eight lesion classes and follow the same YOLO annotation format (`id cx cy w h`), integrating HS-CMU-V2 required no taxonomic changes. The added

Table 3.3: Class-wise statistics of the HS-CMU-V2 dataset (computed from released YOLO labels).

Label (ID)	SM (0)	EC (1)	EP (2)	EPH (3)	EH (4)	IFB (5)	CP (6)	AHE (7)
Images num	79	3	1014	15	5	59	31	23
Bounding box num	79	3	1026	15	5	59	34	24

* Total images: 2,405; empty label files: 1,184; non-empty label files: 1,221; total annotated boxes: 1,245.
* SM, submucous myoma; EC, endometrial cancer; EP, endometrial polyp; EPH, endometrial polypoid hyperplasia; EH, endometrial hyperplasia without atypia; IFB, intrauterine foreign body; CP, cervical polyp; AHE, atypical hyperplasia of the endometrium.

samples introduce greater visual variability and a larger proportion of non-lesion background frames, which is valuable for reducing false-positive detections in clinically ambiguous cases. In practice, HS-CMU-V2 was passed as a second dataset root fed into the same fold-based training pipeline, with unique sample identifiers assigned to prevent filename conflicts between the two sources. Because per-class statistics for the update are not reported with the same detail as the initial release table, potential distribution shift in the added subset is acknowledged.

3.2.3 Additional External Clinical Material

Additional hysteroscopic video material was provided by Professor Vasilios Tanos, Professor of Obstetrics and Gynaecology at the University of Nicosia Medical School, as external clinical data. This material was used exclusively for qualitative demonstration and exploratory analysis of model behavior on external videos with different acquisition and case characteristics compared with the training datasets. Although some videos included lesion patterns represented in the training taxonomy, many findings were outside the trained class scope. Since these videos were not accompanied by frame-level bounding-box annotations, they were not used for supervised training or for quantitative benchmark evaluation under the protocol of this dissertation. Accordingly, all primary quantitative results reported in this implementation are derived only from the annotated HS-CMU and HS-CMU-V2 protocol-defined splits.

3.2.4 Label Taxonomy and YOLO Format

All annotations were prepared in standard YOLO detection format, where each image has a corresponding text file and each annotation row is encoded as

$$\texttt{class\_id x\_center y\_center width height},$$

with normalized coordinates in the range $[0, 1]$.

The label taxonomy used in this dissertation contains eight lesion classes as mentioned before: `SM, EC, EP, EPH, EH, IFB, CP, AHE`. These classes were mapped to integer IDs in the training pipeline configuration.The class-ID mapping is: SM(0), EC(1), EP(2), EPH(3), EH(4), IFB(5), CP(6), and AHE(7), consistent with the dataset specification.

To avoid filename collisions when combining multiple dataset roots, unique sample identifiers of the form `R{root_id}_{stem}` were assigned during data preparation. Empty label files were preserved as valid negative samples, meaning that no lesion bounding box is present in the image. Annotation consistency checks were applied before training to ensure valid image-label pairing across all folds.

3.3 Model Configuration and Training Strategy

3.3.1 YOLO11 Algorithm and Architecture

YOLO11 is the core one-stage detector used in this implementation for lesion localization and recognition. As in the YOLO family, localization and classification are performed in a single forward pass, producing bounding boxes, class labels, and confidence scores directly from network outputs [6, 19]. This design is suitable for high-throughput image and video pipelines where low-latency inference is required.

Algorithmic workflow: For each input image, YOLO11 performs detection in three stages: (i) hierarchical feature extraction in the backbone, (ii) multi-scale feature fusion in the neck, and (iii) dense prediction in the detection head. Final detections are obtained through confidence filtering and non-maximum suppression (NMS), which removes redundant overlapping boxes [2, 19].

Architecture elements: Recent YOLO11 technical descriptions report backbone and neck refinements, including modules such as C3k2, SPPF, and C2PSA, with the aim of improving feature representation and scale-aware detection [3, 17]. These properties are important for hysteroscopic data, where lesions vary in size, texture, and boundary clarity across frames.

The figure 3.1 summarizes the high-level flow of the model and the placement of the key modules discussed next, namely C3k2, SPPF, and C2PSA.

C3k2 block is a lightweight CSP-style feature-extraction block used in YOLO11 backbones/necks to improve the accuracy-efficiency trade-off. Conceptually, it follows Cross-Stage Partial design principles, where part of the feature stream bypasses heavy transformation and is later fused, reducing redundant computation while preserving representation quality [2, 3, 22].

SPPF block is the fast pyramid-pooling module used in YOLO-family architectures to enlarge effective receptive field at low overhead. It is an efficient engineering variant of the spatial

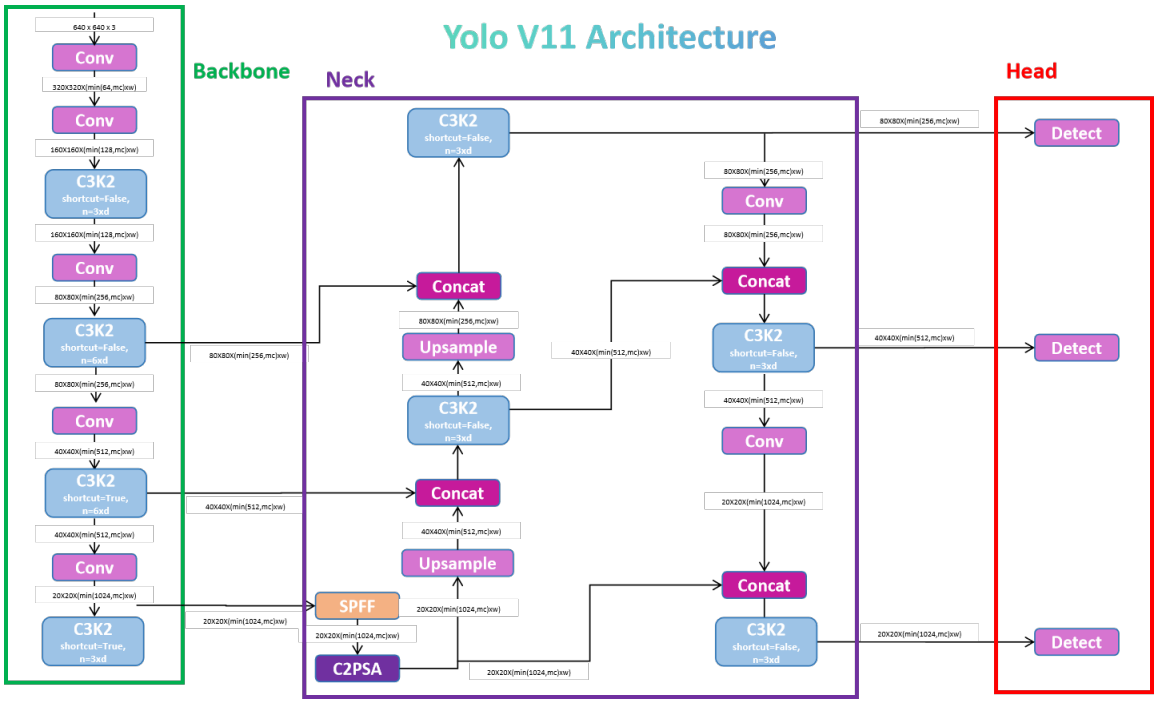

Figure 3.1: Recreated overview of the YOLO11 architecture Backbone, Neck, and Head, based on the YOLO11 architecture description and module flow reported in [2, 3].

pyramid pooling idea, which captures multi-scale context through pooled features at different receptive fields [2, 23].

C2PSA block is an attention-enhanced CSP-style module reported in YOLO11 technical descriptions. Its role is to improve focus on informative channels/spatial regions and strengthen feature selection under cluttered backgrounds and scale variation [2,3]. Since YOLO11 is recent, detailed module-level internals are mainly documented in official technical resources rather than a standalone peer-reviewed architecture paper.

In the implemented `YOLO11m` architecture, the network is organized into 232 layers with approximately 20.06 million parameters and 68.2 GFLOPs. The early backbone stages are formed by sequential `Conv` and `C3k2` modules that progressively increase channel capacity while preserving efficient feature reuse. Mid-level representation is strengthened through `SPPF`, which expands receptive-field context, and `C2PSA`, which enhances channel/spatial attention to informative regions.

The neck follows a multi-scale fusion pattern based on repeated `Upsample` and `Concat` operations, interleaved with `C3k2` refinement blocks, allowing aggregation of fine and coarse semantics. The detection head (`Detect`) operates on three fused feature scales with channel dimensions $[256, 512, 512]$, producing class confidence and box localization outputs for the 8 lesion categories.

YOLO11 variants and their practical differences. The YOLO11 family provides scaled variants (`n, s, m, l, x`) that share the same detection principle but differ in model capacity and computational demand. In practice, the variants trade off:

1. Complexity: larger variants have higher parameter count and FLOPs.
2. Speed: smaller variants provide lower inference latency.
3. Accuracy: larger variants generally reach higher mAP.

Thus, `n,s` are better for strict real-time or low-resource settings, while `l,x` are better when maximum accuracy is prioritized [2, 13].

Table 3.4: YOLO11 detection variants on COCO val2017 (imgsz=640), including latency, parameters, and FLOPs [2].

Variant	mAP^{val}_{50-95}	CPU ONNX (ms)	T4 TensorRT10 (ms)	Params (M)	FLOPs (B)
YOLO11n	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

In this implementation, compute cost is quantified by parameter count and FLOPs (Table 3.4) and jointly considered with latency and mAP during model selection.

The `YOLO11m` variant was selected because it offers a balanced operating point between lightweight and high-capacity models. Official COCO benchmark values show that `YOLO11m` improves detection accuracy over `n,s` while keeping latency and complexity clearly lower than `l,x` [2]. This selection supports the requirements of the present implementation:

1. sufficient capacity for subtle lesion appearance patterns,
2. feasible training cost under cross-validation,
3. practical inference speed for image and video analysis.

Compared with earlier YOLO generations, YOLO11 is reported to improve the balance between accuracy, speed, and computational efficiency while preserving the operational simplicity of one-stage detection [2, 3]. Official benchmark tables also indicate stronger COCO detection performance than corresponding YOLOv8 variants at competitive model complexity for multiple scales [2].

Latency vs. Accuracy Interpretation Figure 3.2: The X-axis represents inference latency in milliseconds per image with T4 TensorRT10 FP16. Lower values indicate faster inference. The Y-axis represents COCO mAP_{50-95}, which is average detection precision across IoU thresholds from 0.50 to 0.95. Higher values indicate better detection quality. The observed trend

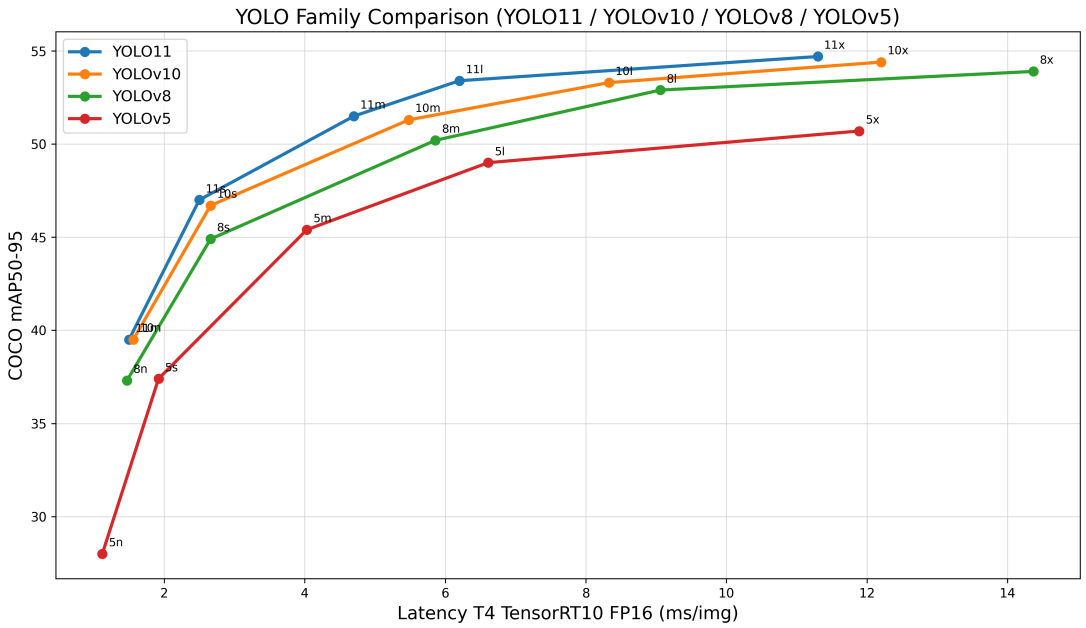

Figure 3.2: Latency-accuracy comparison across YOLO11, YOLOv10, YOLOv8, and YOLOv5 on COCO mAP_{50-95} versus T4 TensorRT10 FP16 latency (ms/img), reproduced from official Ultralytics chart data [2].

shows a standard speed-accuracy trade-off. Lightweight models are faster but usually less accurate, while larger models are more accurate but slower. The YOLO11 curve lies on a favorable latency-accuracy frontier relative to YOLOv10, YOLOv8, and YOLOv5, indicating stronger accuracy at comparable latency levels for multiple scales. The point labels `n/s/m/l/x` denote model scales as nano, small, medium, large, and extra-large within each family.

Evaluation dimensions used for model selection: Performance is measured with mAP_{50-95}. Speed is measured with inference latency in milliseconds per image, and equivalently with frames per second after conversion. Compute cost is represented by model size and complexity, reported through parameter count and FLOPs.

3.3.2 Baseline Training Configuration

The baseline configuration defines the reference experimental setup against which all later parameter changes and ablation studies are compared. The objective is to ensure methodological consistency, reproducibility, and clear attribution of performance differences to controlled modifications rather than to uncontrolled training variability.

Model, task, and data protocol. The baseline detector is `YOLO11m` (`yolo11m.pt`) trained in object-detection mode using the Ultralytics framework [2]. Training data are drawn from HS-CMU and HS-CMU-V2, using the unified 8-class YOLO label taxonomy described in Section 3.2.4 [1, 12]. A 5-fold cross-validation protocol is used (`NUM_FOLDS=5`, `NUM_RUNS=1`,

RANDOM_SEED=42), with fold-specific train/val/test splits and generated data.yaml files.

Optimization setup and rationale for AdamW. The baseline optimizer is AdamW with initial learning rate lr0 $= 5x10^{-4}$. AdamW was selected because it decouples weight decay from the adaptive gradient update, which generally yields more stable regularization behavior than coupling L2 penalty directly into Adam's moment-based update [24,25]. Formally, Adam computes first- and second-moment estimates:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$$

with bias correction:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}.$$

AdamW applies a decoupled shrinkage term:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \eta \lambda \theta_t,$$

where η is learning rate and λ is weight-decay strength [25].

Table 3.5: Baseline training configuration

Parameter	**Value**
Model	`yolo11m.pt`
Image size (`imgsz`)	640
Batch size (`batch`)	16
Epochs (`epochs`)	200
Optimizer	AdamW
Initial learning rate (`lr0`)	5×10^{-4}
Cosine LR schedule (`cos_lr`)	True
Early stopping patience (`patience`)	50
Workers (`workers`)	4
Mixed precision (`amp`)	True
Cache mode (`cache`)	disk
Loss weights	`box=7.5`, `cls=1.0`, `dfl=1.5`
Freeze (`freeze`)	0
Mosaic (`mosaic`)	0.0
Close mosaic (`close_mosaic`)	0
MixUp (`mixup`)	0.0
Scale augmentation (`scale`)	0.0
Empty-negative ratio (`EMPTY_TO_POS_RATIO`)	0.2

The baseline settings in Table 3.5 control how the model sees data, how it learns, and how strongly different training objectives are enforced. The parameter `imgsz` sets the input image size used during training and inference. The parameter `batch` sets how many images are processed before each weight update. Together, these two values affect memory usage, training stability, and processing speed.

Learning behavior is defined mainly by `optimizer`, `lr0`, `cos_lr`, and `patience`. The `optimizer` specifies the update method used for model weights, in this case AdamW. The parameter `lr0` sets the initial learning rate, which controls how large each update step is at the beginning of training. The parameter `cos_lr` activates cosine learning-rate scheduling so the learning rate gradually decreases in a smooth way during training. The parameter `patience` sets how many epochs without validation improvement are tolerated before early stopping.

Execution behavior is controlled by `epochs`, `workers`, `amp`, and `cache`. The parameter `epochs` defines the maximum number of full training passes over the dataset. The parameter `workers`

sets how many parallel processes load data. The option `amp=True` enables mixed-precision computation, which usually reduces memory consumption and improves speed. The setting `cache=disk` stores cached data on disk to reduce repeated loading overhead.

The training objective is balanced by `box`, `cls`, and `dfl`. The parameter `box` controls the importance of bounding-box localization accuracy. The parameter `cls` controls the importance of class prediction accuracy. The parameter `dfl` controls the importance of fine-grained box regression through distribution focal loss. These weights determine how much each objective contributes to total loss during optimization.

Data augmentation is controlled by `mosaic`, `close_mosaic`, `mixup`, and `scale`. The parameter `mosaic` combines multiple training images into one composite sample to increase visual variation. The parameter `close_mosaic` disables mosaic during the final part of training to improve alignment with natural image appearance. The parameter `mixup` blends pairs of images and labels to improve regularization. The parameter `scale` applies random resizing to make the model more robust to object-size variation. In E0 these augmentation controls are set to zero, which defines the reference condition without mosaic, mixup, or scale perturbation.

The parameter `EMPTY_TO_POS_RATIO` controls the inclusion of empty frames during training, where empty frames are frames without lesion labels. This value sets how many such empty-label negative samples are added relative to positive samples.

Execution environment and reproducibility: Baseline runs are executed on a High Performance Computing environment using one NVIDIA V100 GPU, CUDA 11.7, Python 3.10, and SLURM job orchestration. Reproducibility is ensured through fixed seeds, fold-specific artifacts such as `results.csv` and best-checkpoint weights, and script-level configuration versioning.

Validation outputs: For each fold, the selected checkpoint `best.pt` is evaluated on validation and test subsets. The primary reported detection metrics are precision, recall, mAP@0.5, and mAP@0.5:0.95, followed by fold aggregation with mean and standard deviation in Section 3.4.

3.3.3 Parameter Changes and Ablation Settings

After establishing the baseline configuration (Section 3.3.2), a set of controlled parameter changes was explored to evaluate sensitivity to input scale, augmentation policy, and preprocessing strategy. The objective was to identify configurations that improve detection quality without disproportionate computational cost.

Ablation design principle: Each ablation changes one configuration component at a time while keeping the remaining baseline settings fixed. This isolates the effect of each modification and supports causal interpretation of metric changes.

Ablation groups: The explored settings are organized into three groups. The first group is input

scale, which includes comparisons across multiple input resolutions such as 640 and 1280. The second group is augmentation and sampling policy, which examines settings with and without mosaic-driven augmentation and associated training controls, together with different empty-label negative sampling ratios. The third group is optimization and model-capacity settings, which covers adjustments of learning rate, selected loss weights, and detector scale through smaller and larger YOLO11 variants.

Reporting format: For each ablation, fold-level validation and test metrics including mAP@0.5, mAP@0.5:0.95, precision, and recall are recorded and aggregated by mean and standard deviation. Comparisons are interpreted jointly with latency and compute indicators to preserve a balanced performance view.

Reproducibility note: Exact training arguments for each run were retrieved from saved experiment artifacts, including fold-level `args.yaml`, `results.csv`, and exported metric summaries, to ensure consistency between reported configurations and executed experiments.

3.4 Evaluation Metrics and Validation Protocol

This section defines the evaluation criteria and validation procedure used to assess detection performance. Evaluation is performed with the Ultralytics validation interface (`model.val()`) across all folds and splits, ensuring consistent metric computation, IoU handling, and confidence-threshold processing [2]. The protocol combines standard object detection metrics with fold-wise aggregation to provide robust and reproducible estimates of model quality.

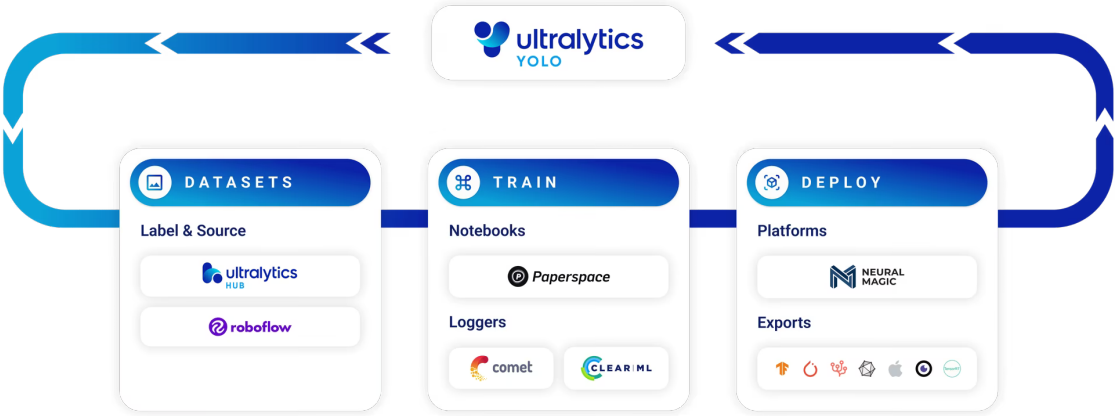

Figure 3.3: Ultralytics workflow overview dataset preparation, training, and deployment adapted from official documentation [4].

3.4.1 Core Detection Metrics

Validation outputs from `model.val()`: For each fold, the trained checkpoint is evaluated on both `val` and `test` subsets. Evaluation is performed with `best_model.val` using `split="val"` and `split="test"`. The primary aggregate outputs used in this implementation are mean precision `box.mp`, mean recall `box.mr`, mAP@0.5 `box.map50`, and mAP@0.5:0.95 `box.map` [2].

Per-class YOLO validation metrics: Alongside global aggregate scores, the Ultralytics validation report provides per-class statistics that support detailed error analysis and imbalance-aware interpretation. The class field indicates the lesion category being evaluated, such as SM, EC, EP, EPH, EH, IFB, CP, and AHE. The image field reports how many validation images include at least one instance of that class, while the instance field reports the total number of annotated objects for that class. Precision `P` expresses the proportion of predicted positives that are correct, and recall `R` expresses the proportion of ground-truth objects that are successfully detected. The metric `mAP@0.5` summarizes detection quality at IoU 0.50, whereas `mAP@0.5:0.95` summarizes localization performance over IoU thresholds from 0.50 to 0.95 with step 0.05, giving a stricter overall estimate.

These per-class metrics complement global fold-level summaries and are particularly important in medically imbalanced datasets, where overall averages may mask poor performance in rare but clinically relevant classes [6, 26, 27].

Each predicted box is evaluated against ground truth using IoU:

$$\text{IoU} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|},$$

where B_p is the predicted box and B_{gt} is the ground-truth box [6, 27].

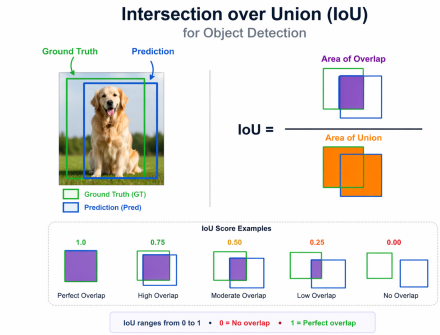

Figure 3.4: Visual interpretation of Intersection over Union (IoU): overlap area divided by union area between predicted and ground-truth bounding boxes, adapted from [5].

Based on matching and IoU thresholding, a true positive `TP` denotes a correct detection of an existing object, a false positive `FP` denotes an incorrect detection due to wrong class assignment,

poor localization, or duplicate prediction, and a false negative `FN` denotes a missed ground-truth object.

Precision, Recall, AP, and mAP are computed from these counts:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}.$$

Average Precision (AP) is the area under the precision-recall curve per class, and mean Average Precision (mAP) is:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i,$$

where N is the number of classes [26, 27].

mAP@0.5 is used as the primary checkpoint-selection indicator in the global model-selection step across folds, while mAP@0.5:0.95 is reported as a stricter secondary localization metric.

3.4.2 Validation Curves and Plots

In addition to scalar metrics, `model.val()` in Ultralytics generates confidence-dependent curves that are used to interpret operating-point behavior.

BoxP curve: This curve shows precision versus confidence threshold.

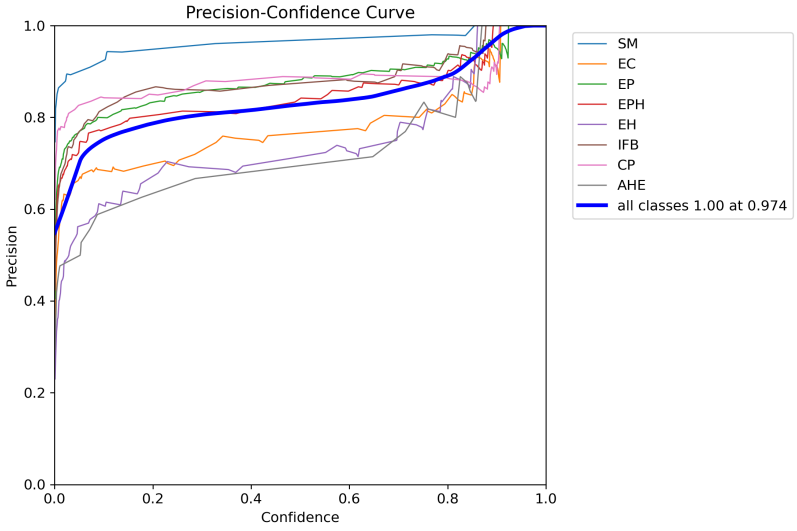

Figure 3.5: Precision-confidence curve BoxP generated by `model.val()`.

BoxR curve: This curve shows recall versus confidence threshold.

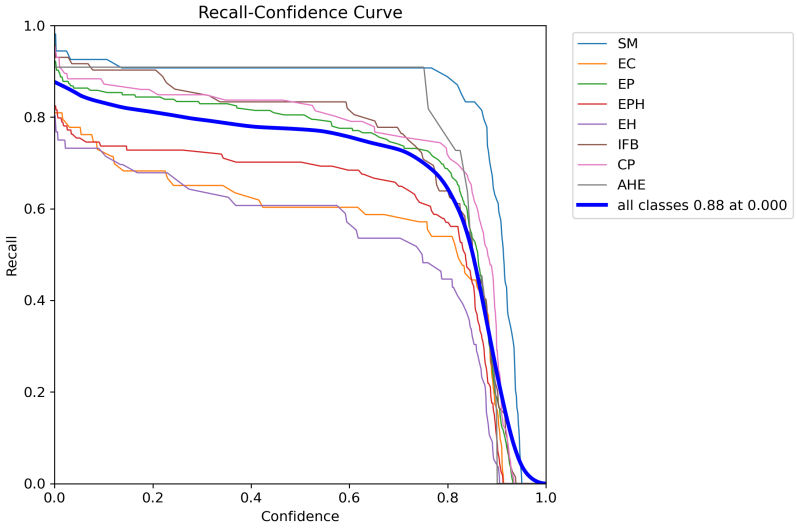

Figure 3.6: Recall-confidence curve BoxR generated by `model.val()`.

BoxF1 curve: This curve shows F1 score versus confidence threshold.

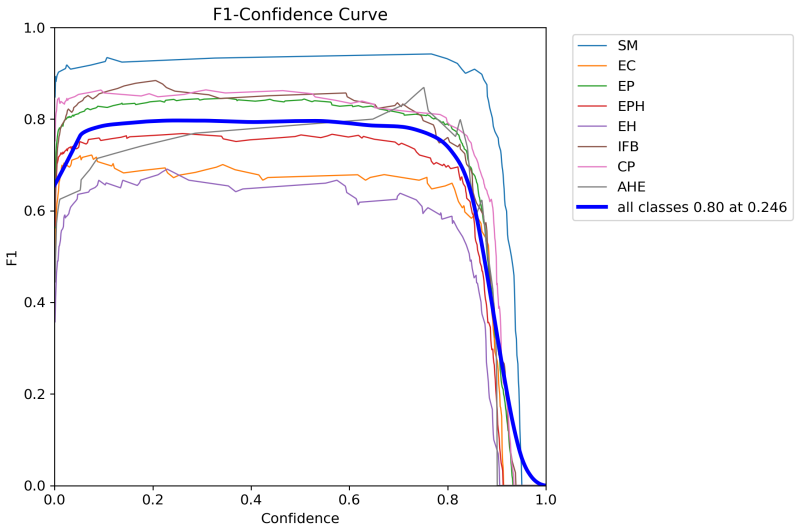

Figure 3.7: F1-confidence curve BoxF1 generated by `model.val()`.

The F1 score is defined as:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

These curves are used for threshold diagnostics and practical deployment tuning. Higher confidence thresholds usually increase precision but reduce recall. Lower confidence thresholds usually increase recall but may increase false positives. The F1 peak provides a balanced operating point.

BoxPR curve: This curve shows the precision-recall trade-off over confidence thresholds.

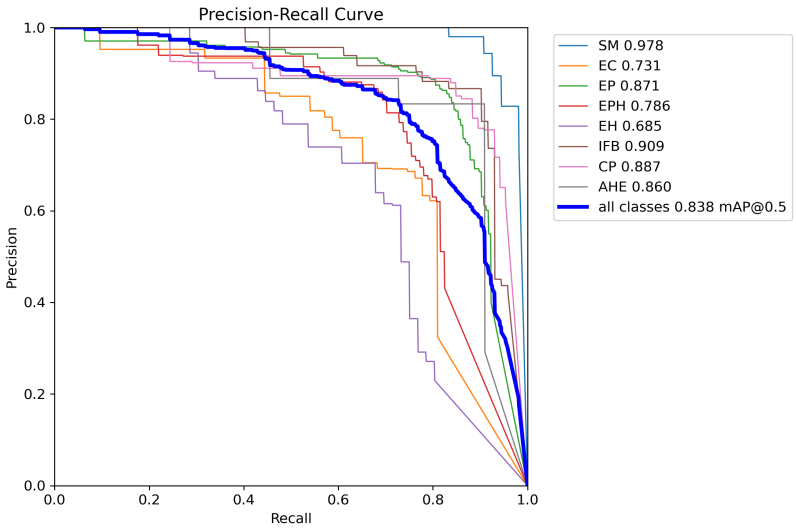

Figure 3.8: Precision-recall curve BoxPR generated by `model.val()`.

Confusion-matrix diagnostics.

`confusion_matrix.png`: This plot shows class-wise confusion in absolute counts.

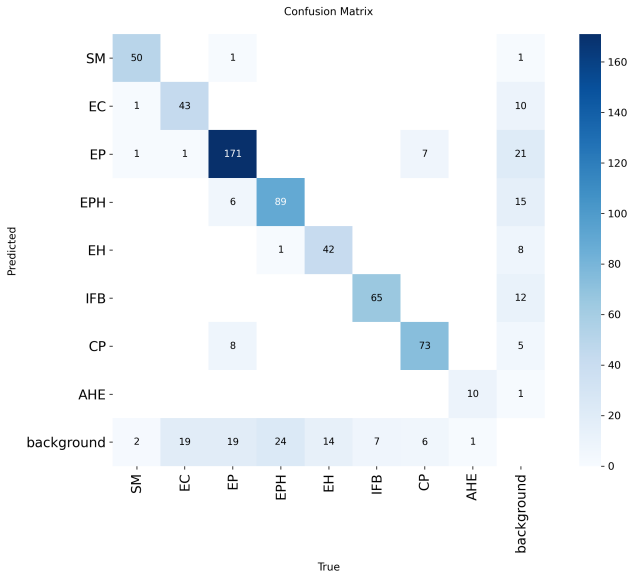

Figure 3.9: Confusion matrix with absolute counts generated by `model.val()`.

`confusion_matrix_normalized.png`: This plot shows class-wise confusion in normalized proportions and is useful under class imbalance.

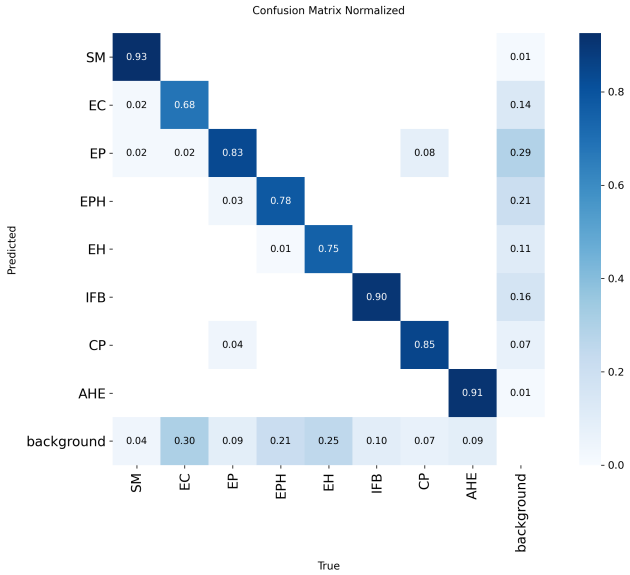

Figure 3.10: Normalized confusion matrix with class-wise proportions generated by `model.val()`.

Batch-level qualitative diagnostics.

`val_batch*_labels.jpg`: These images show sampled validation frames with ground-truth boxes.

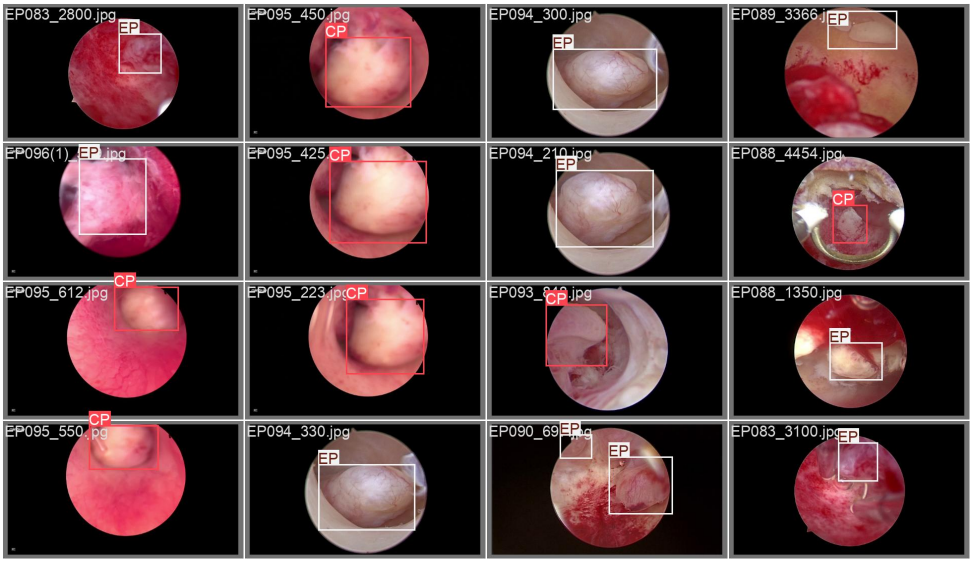

Figure 3.11: Sampled validation images with ground-truth boxes from `val_batch0_labels.jpg`.

`val_batch*_pred.jpg`: These images show corresponding model predictions, including boxes, classes, and confidences, for direct visual comparison.

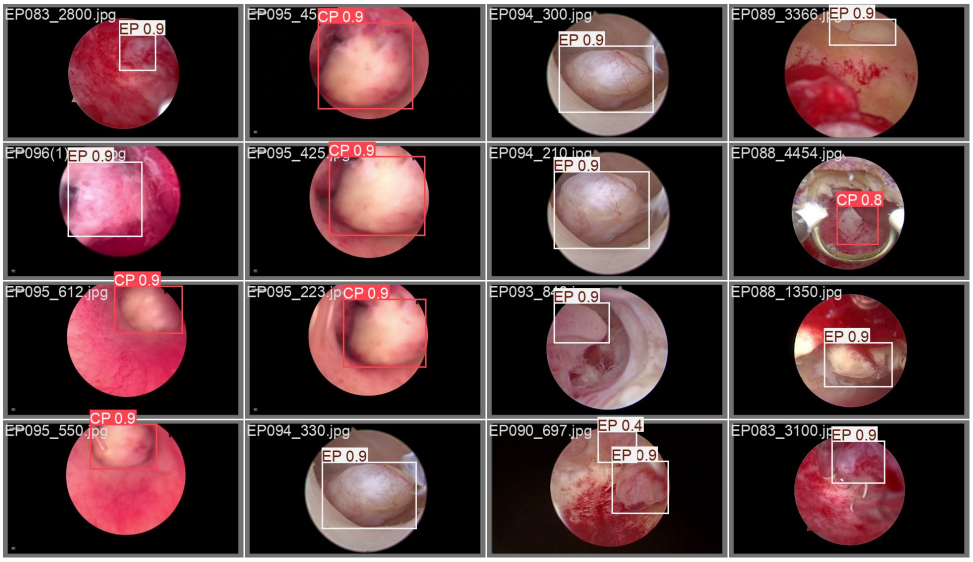

Figure 3.12: Sampled validation predictions from `val_batch0_pred.jpg`.

This analysis is useful in clinically sensitive settings where false negatives and false positives have different operational costs [2].

3.4.3 Checkpoint Selection and Fold Aggregation

Cross-validation structure: The implementation uses 5-fold cross-validation with `NUM_FOLDS=5` and `StratifiedGroupKFold`. In each fold, the outer split assigns approximately 80% of the data to a training pool and 20% to a held-out test fold. The training pool is then internally split with `train_test_split(test_size=0.2)`. This yields effective proportions of approximately 64% train, 16% validation, and 20% test.

Checkpoint policy: Within each fold, `best.pt` is selected by Ultralytics validation fitness during training. Across folds, the implementation selects the global best model using validation mAP@0.5 as the primary criterion.

Fold-level metrics and final reporting: For each fold, validation and test metrics are recorded, including P, R, mAP@0.5, and mAP@0.5:0.95. Final results are aggregated as mean and standard deviation across folds. The mean provides the central performance estimate, while the standard deviation quantifies stability and split-to-split variability.

This protocol provides both accuracy and robustness evidence for comparing model configurations under consistent evaluation conditions.

3.5 Summary

This chapter defined the methodological foundation of the study, including model selection logic, dataset integration protocol, baseline training setup, and evaluation strategy. The framework was structured to ensure consistency across folds, reproducibility of settings, and comparability of results across different configurations.

The chapter also established the metric hierarchy used in subsequent analysis, with mAP@0.5 as the primary checkpoint-selection indicator and mAP@0.5:0.95, precision, and recall as complementary quality measures. Together with confidence-threshold diagnostics and fold aggregation, this protocol provides a consistent basis for the experimental results presented in the next chapter.

In conclusion, the methodology and evaluation design presented for YOLO11 provide a rigorous and practically oriented foundation for assessing detector effectiveness. These metrics and validation procedures are essential for identifying strengths and limitations of each configuration, guiding model refinement, and supporting evidence-based conclusions regarding performance in realistic deployment-oriented settings.

4 Experimental Results

4.1 Experimental Setup for Evaluation

This chapter presents the experimental findings of the proposed hysteroscopic lesion-detection pipeline. Results are reported objectively, while interpretation is deferred to Chapter 5.

The primary detector is YOLO11m, evaluated with fold-based validation under a fixed protocol. Performance is reported with precision, recall, mAP@0.5, and mAP@0.5:0.95, which are standard metrics in object-detection evaluation [6, 26, 27].

The experiment set includes baseline and controlled parameter variations (E0, E1, E2, E4, E5, E6, E7, E9), with RetinaNet-ResNet50 evaluated as an additional comparison baseline.

Table 4.1: Experimental protocol summary (as implemented in the training pipeline).

Item	Configuration
Primary detector	YOLO11m (`yolo11m.pt`)
Comparison detector	RetinaNet-ResNet50
Number of runs	`NUM_RUNS = 1`
Number of folds	`NUM_FOLDS = 5`
Random seed	`RANDOM_SEED = 42`
Split strategy	StratifiedGroupKFold + train/val split (`test_size=0.2`)
Grouping key	Sample identifier (`sid`)
Core metrics	Precision, Recall, mAP@0.5, mAP@0.5:0.95
Dataset roots	HS-CMU and HS-CMU-V2
Negative sampling	Train-only empty-label downsampling (`EMPTY_TO_POS_RATIO`)
Checkpoint selection	Best checkpoint selected by highest validation mAP@0.5
Reported statistics	Fold-wise metrics reported as mean $\pm$ std

Implementation traceability All protocol and environment values reported in this section are extracted from the training pipeline and SLURM job scripts Exact excerpts are provided in Appendix A

4.1.1 Computing Environment

All experiments were executed on the University HPC cluster under SLURM scheduling. The software environment was initialized through module loading and virtual-environment activation in each batch script.

The experiments were executed on `turing1.ucy.ac.cy`. The CPU platform was an Intel Xeon Gold 6226R at 2.90GHz, with 32 CPUs across 2 sockets, 16 cores per socket, and 1 thread per core. System memory was 92 GB. GPU acceleration used a Tesla V100-SXM2-32GB. The software stack consisted of Python 3.10.8, PyTorch 2.0.1+cu117, Ultralytics 8.4.9, and CUDA toolkit 11.7. Job orchestration was managed by SLURM with `sched/backfill`, `select/cons_res`, and `CR_CORE_MEMORY`.

4.2 Endoscopy Imaging Detection Results

4.2.1 Introduction

In this subsection, I present the outcomes of deploying the trained YOLO11m model for lesion detection in hysteroscopic images. The aim of these experiments was to evaluate the model's ability to accurately identify and localize abnormal regions of interest across the predefined clinical classes, using the HS-CMU and HS-CMU-V2 datasets described in the previous chapter. The relevance of this task is directly linked to the need for early and reliable identification of intrauterine abnormalities, which can support timely clinical decision-making.

The following results are reported through standard object-detection metrics and qualitative visual outputs. Beyond numerical performance, these findings provide evidence on the practical behavior of the model under realistic image variability. The full workflow, from dataset preparation and fold-based training to validation and test evaluation, was executed under a fixed protocol to ensure methodological consistency and reproducibility.

4.2.2 Methodology Recap

In progressing to the experimental phase, I briefly recapitulate the training procedure that underpins the detection results. The core of this process was the application of a YOLO11m detector on hysteroscopic image data collected from HS-CMU and HS-CMU-V2, under the fixed protocol defined in Chapter 3.

A concise summary of the implementation is given below.

The required modules were first imported, including the Ultralytics framework, which provides the training and evaluation interface for YOLO-based detectors. The YOLO model was then initialized from pretrained weights `yolo11m.pt` and configured for object detection. For each fold, a fold-specific `data.yaml` file was generated to define dataset paths, number of classes, and class names used during training and evaluation. Model training was executed with fixed experiment settings, including image size, optimizer, learning rate, batch size, and augmentation configuration. The best checkpoint was selected according to validation mAP@0.5 and then evaluated on validation and test splits using precision, recall, mAP@0.5, and mAP@0.5:0.95.

A representative code segment from the training pipeline is shown below.

```
from ultralytics import YOLO

YOLO_MODEL = "yolo11m.pt"
EPOCHS = 200
```

```
OPTIMIZER = "AdamW"

# IMAGE_SIZE, BATCH_SIZE, LEARNING_RATE, and selected augmentation
# settings were configured per experiment

data_yaml = os.path.join(fold_dir, "data.yaml")
with open(data_yaml, "w") as f:
    f.write(
        f"path: {fold_dir}\ntrain: train/images\nval: val/images\n"
        f"test: test/images\nnc: {len(class_names)}\nnames: {class_names}\n"
    )

model = YOLO(YOLO_MODEL)
results = model.train(data=data_yaml, task="detect", imgsz=IMAGE_SIZE,
                      epochs=EPOCHS, batch=BATCH_SIZE, lr0=LEARNING_RATE,
                      optimizer=OPTIMIZER, device=f"cuda:{GPU_ID}")
```

This code segment captures the central training logic used throughout the experiments. After training, the framework produces epoch-wise curves and evaluation plots, which are presented in the following subsection to analyze convergence, generalization, and detection quality.

4.2.3 Training and Evaluation Curves

The YOLO training outputs include epoch-wise loss curves and metric curves, used to assess convergence and generalization.

Interpretation of training curves. For clarity, each panel in Figure 4.1 is interpreted separately below.

Top row: training losses and core detection metrics. The metric `train/box_loss` measures bounding-box regression error on training data, and a decreasing trend indicates improved localization. The metric `train/cls_loss` measures training classification error, and lower values indicate better class discrimination. The metric `train/dfl_loss` captures fine-grained box-boundary refinement during training, and a downward trend indicates improved geometric precision. The metric `metrics/precision(B)` is the fraction of predicted lesion detections that are correct. The metric `metrics/recall(B)` is the fraction of true lesion instances that are successfully detected.

Bottom row: validation losses and aggregate accuracy. The metric `val/box_loss` is validation localization error and indicates how well localization generalizes to unseen data. The metric `val/cls_loss` is validation classification error and reflects class-separation generalization.

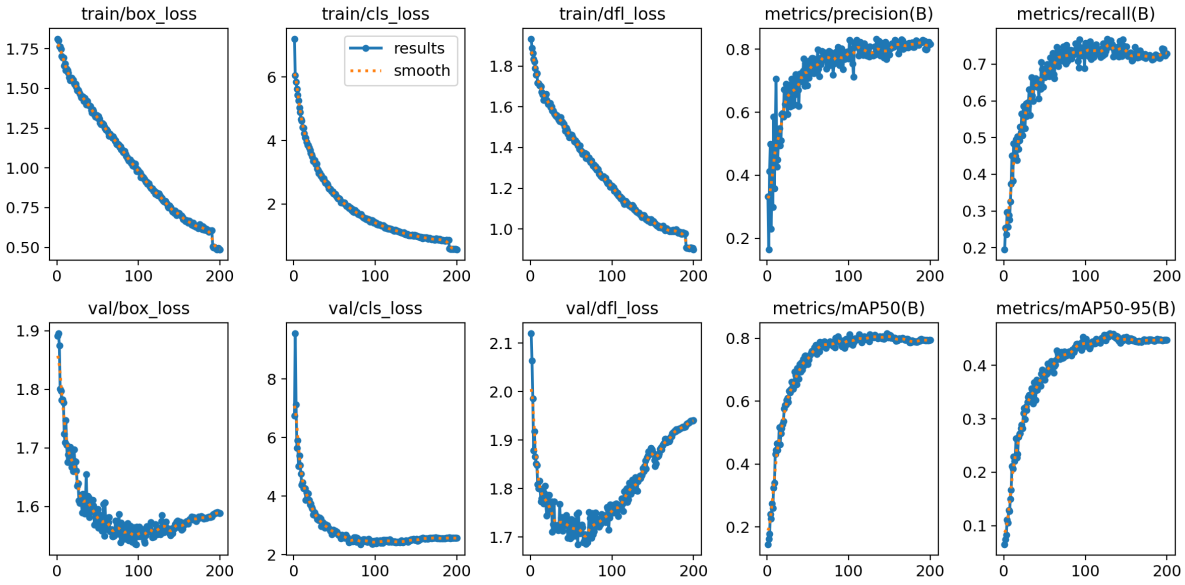

Figure 4.1: Training and validation curves for a representative fold of the selected YOLO11m configuration.

The metric `val/dfl_loss` measures fine bounding-box refinement on validation data. In this run, it decreases early and then rises slightly in later epochs, suggesting mild late-stage instability in fine localization. The metric `metrics/mAP50(B)` is mean Average Precision at IoU 0.50 and summarizes detection performance under a moderate overlap criterion. The metric `metrics/mAP50-95(B)` averages mAP over IoU thresholds 0.50 to 0.95 and is stricter because it also rewards precise localization.

Taken together, these curves are used to assess optimization progress, generalization behavior, and the balance between detection sensitivity and localization precision.

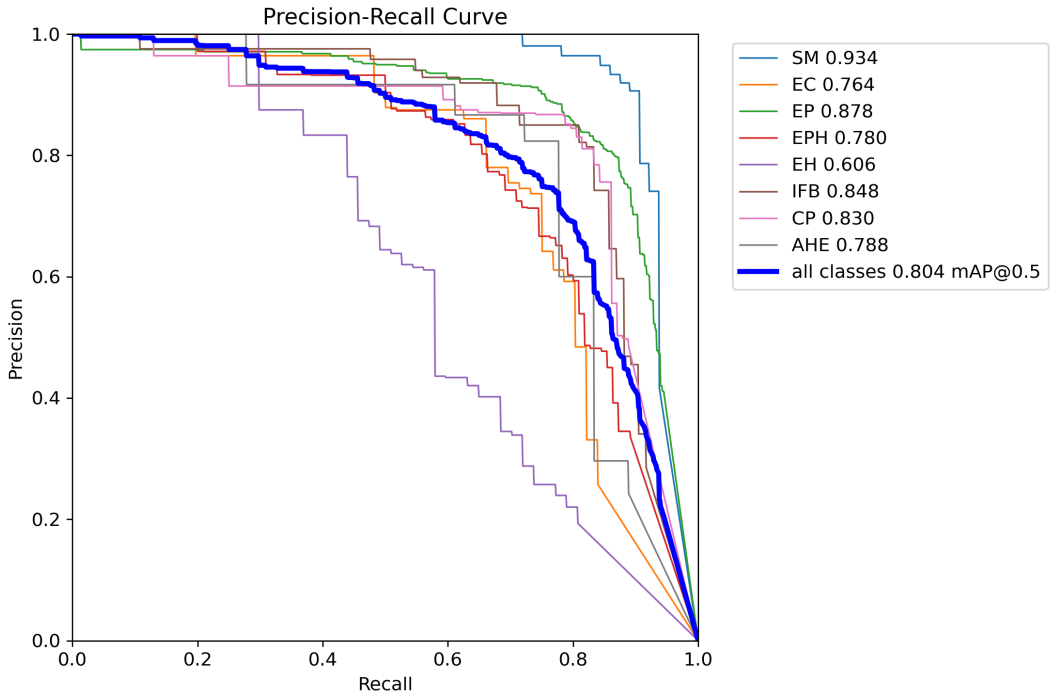

Figure 4.2: Precision-Recall curve for the selected YOLO11m evaluation setting.

Interpretation of Figure 4.2 Precision-Recall Curve. The aggregate curve shown in blue reports `mAP@0.5 = 0.804`. This value indicates strong overall detection performance at IoU 0.50 and confirms that the model can detect a large proportion of lesions while maintaining good precision. The AP values are not uniform across classes. The model performs strongly for `SM (0.934)` and `EP (0.878)`. Performance is lower for `EH (0.606)`, which suggests that this class is more difficult to distinguish and localize under the current configuration. Precision remains high across a broad part of the recall range and then decreases more clearly near very high recall. This behavior is expected when the detector is tuned to recover more positives, since additional true detections are usually accompanied by more false positives. Most classes preserve a useful precision-recall balance over relevant operating points. The `EH` class remains the most challenging in this setting, which is consistent with lower class representation and higher visual variability.

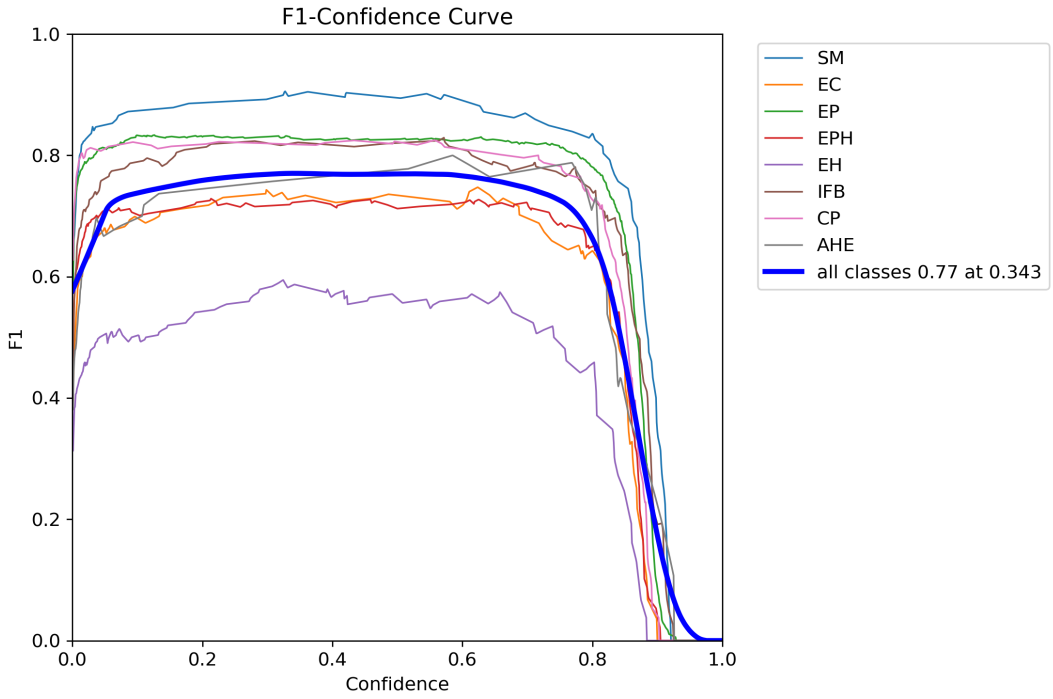

Figure 4.3: F1-score as a function of confidence threshold for the selected YOLO11m evaluation setting.

Interpretation of Figure 4.3 F1-Confidence Curve. The aggregate F1 curve reaches its maximum at `F1 = 0.77` for confidence `0.343`. This point provides the most effective balance between precision and recall for this evaluation setting and can be used as a practical default threshold for reporting fold-level behavior. Around the middle confidence range, the F1 value changes gradually and remains comparatively stable. This indicates that moderate threshold adjustments in this region do not produce large performance fluctuations and suggests a reliable operating zone. For confidence values above about `0.8`, the F1 score drops more steeply. The main reason is reduced recall, since stricter confidence filtering removes not only false positives but also a substantial number of true positive detections. Class-wise curves show consistent differences in difficulty. Stronger classes such as `SM` and `EP` maintain higher F1 across a wider threshold range, while `EH` remains lower across most thresholds, indicating weaker separability under the current configuration.

4.2.4 Confusion Matrix and Class-Level Behaviour

The confusion matrix provides class-level error analysis for the representative fold and complements aggregate metrics such as mAP, precision, and recall. It highlights where predictions are correct and where misclassification patterns are concentrated across lesion categories.

This section uses the test split of 1,158 images. Matrix entries are computed at lesion-instance

level, not image level, so one frame may contribute to multiple outcomes when it contains multiple lesions. This is expected in hysteroscopic data where lesion count varies across frames.

Detection outcomes are defined as follows. A true positive denoted as `TP` is a correctly detected lesion with correct class assignment. A false negative denoted as `FN` is a ground-truth lesion that is missed or not matched as correct. A false positive denoted as `FP` is a predicted lesion that does not match a valid ground-truth lesion.

Empty-label frames are included as hard negatives to improve false positive control. Accordingly, interpretation focuses on lesion detections, missed lesions, and background-related false positives. The absolute and normalized matrices are reported together to show both error volume and class-wise proportions.

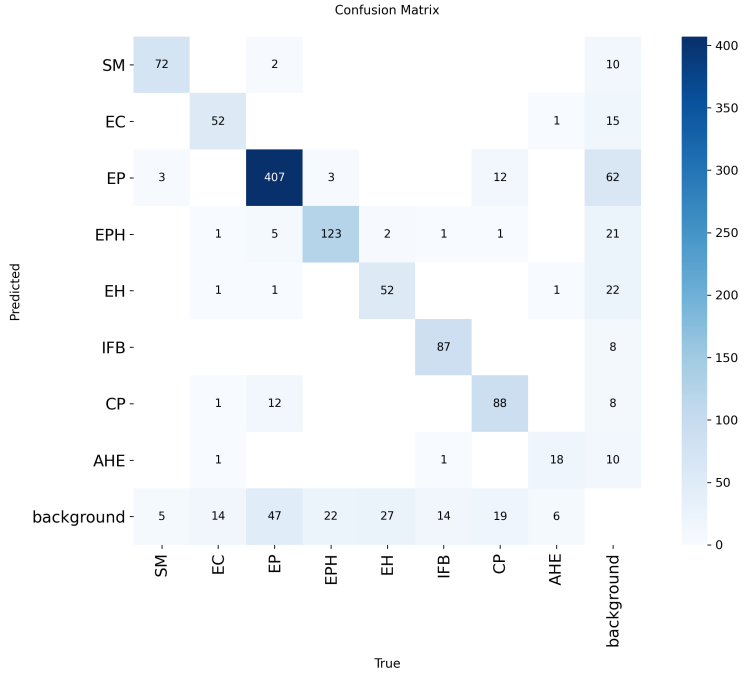

Figure 4.4: Absolute confusion matrix for the representative fold on the test split.

Interpretation of Figure 4.4. The diagonal cells correspond to correct class assignments. Large diagonal counts for EP, EPH, IFB, CP, and SM indicate strong recognition for these classes in the representative fold. EP shows the largest absolute number of correct detections, which is consistent with stronger representation and stable learning for this class. Off-diagonal cells identify where class confusions occur. These entries reveal that some lesions are assigned to neighboring categories with similar visual appearance. The background column contains lesion predictions where the true label is background, which corresponds to false positive detections. The background row contains true lesion instances that were not recovered as correct lesion predictions, which corresponds to missed detections and reduced sensitivity for affected classes.

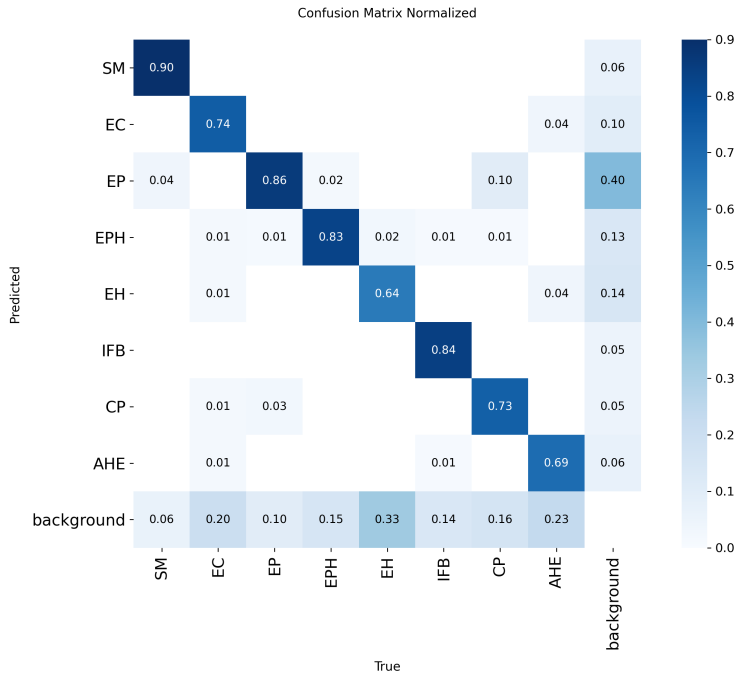

Figure 4.5: Normalized confusion matrix for the representative fold on the test split.

Interpretation of Figure 4.5. The normalized matrix reports per-class proportions and allows fair comparison across classes with different sample sizes. Diagonal proportions are high for SM, EP, EPH, and IFB, indicating strong class retention after normalization. EC, CP, and AHE remain moderate, while EH is lower than most other classes, confirming that EH is relatively harder under the current configuration. Background-linked proportions highlight classes with higher miss tendency and classes more prone to false positive spillover, which is consistent with lower separability in challenging visual conditions. Together with the PR and F1 curves, the normalized matrix confirms that performance is strong for major classes and identifies specific classes where further tuning is most needed.

4.2.5 Visual Aids

The following examples are taken from the validation split and are presented as paired views of annotation and prediction. For each batch, the top panel shows the reference labels and the bottom panel shows the detector output for the same frames. These sample images are included to provide a direct visual comparison of lesion localization performance in practical hysteroscopic scenes.

Batch 1 Visual Comparison EP.

Figure 4.6: Validation batch 1.

Batch 1 description. This batch is centered on EP. The predicted boxes generally align well with the annotated lesion regions, while the associated confidence scores remain high for clear appearances and decrease in frames with weaker boundaries or reduced contrast.

Batch 2 Visual Comparison CP and AHE.

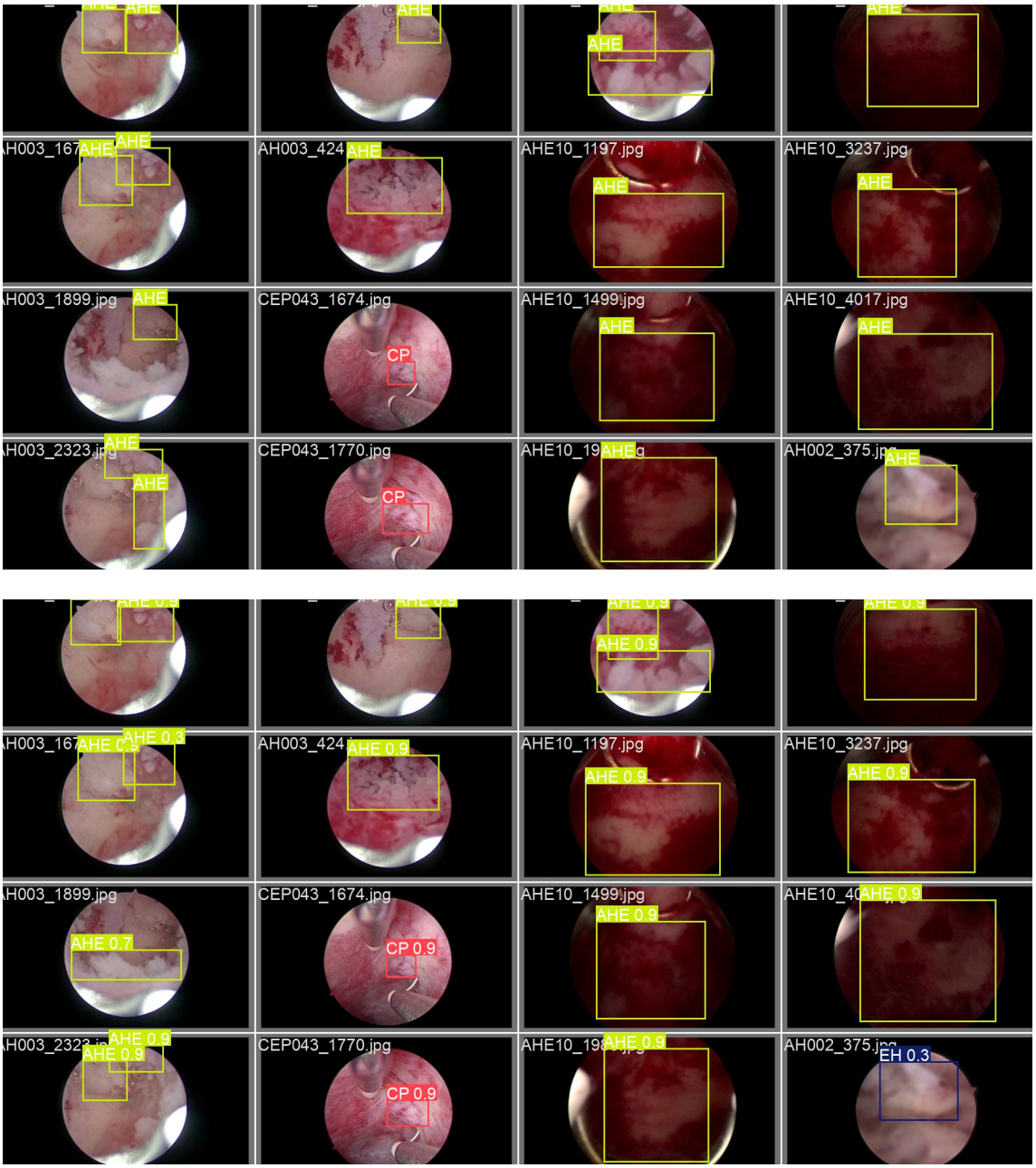

Figure 4.7: Validation batch 2.

Batch 2 description. This batch includes CP and AHE, which are visually more variable. The model still detects the main targets in most frames, but confidence scores show wider fluctuation, reflecting increased difficulty in borderline or heterogeneous lesion patterns.

4.2.6 Feature-Map and Heatmap Visualizations

In addition to quantitative metrics, attention-based visualization is used to examine detector focus on a representative frame. This analysis evaluates whether activation is concentrated on clinically relevant lesion regions and provides qualitative evidence on attention behavior under the trained detection setting.

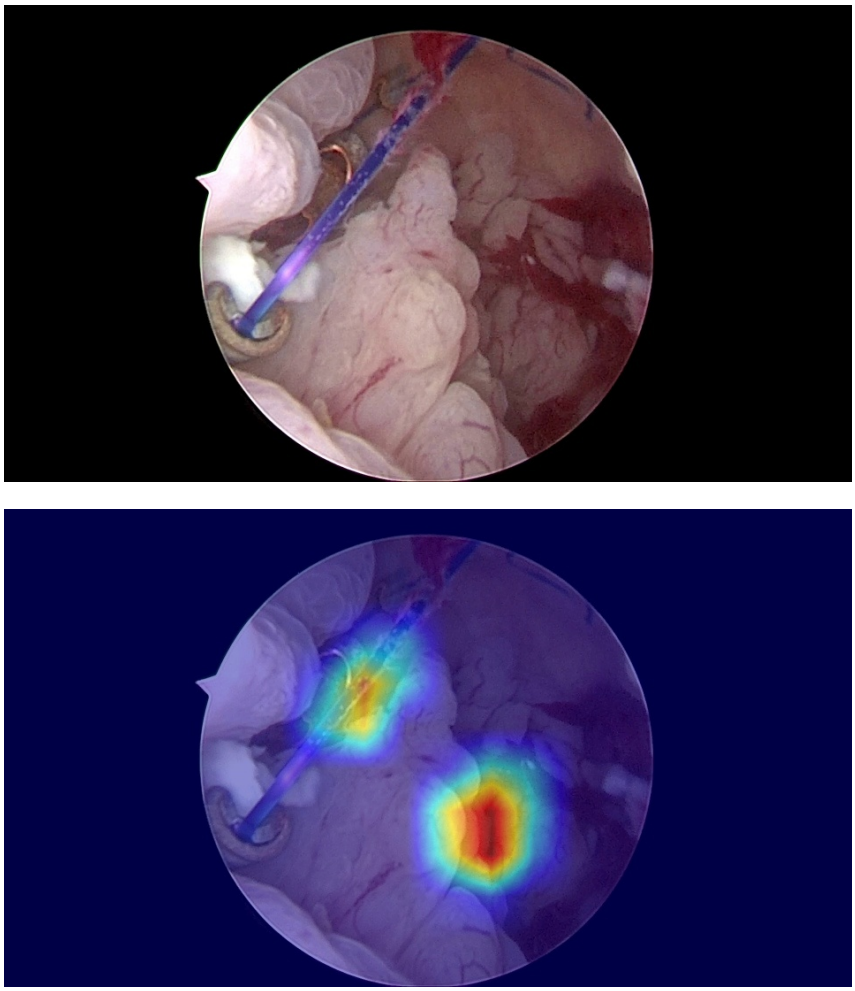

Figure 4.8: Original image (top) and corresponding EigenCAM activation heatmap (bottom).

The activation pattern is spatially focused and non-uniform, with higher responses concentrated in anatomically relevant regions of the hysteroscopic view. This distribution suggests that the detector emphasizes informative lesion-associated tissue cues rather than broad background response, supporting the localization trends observed in the quantitative evaluation.

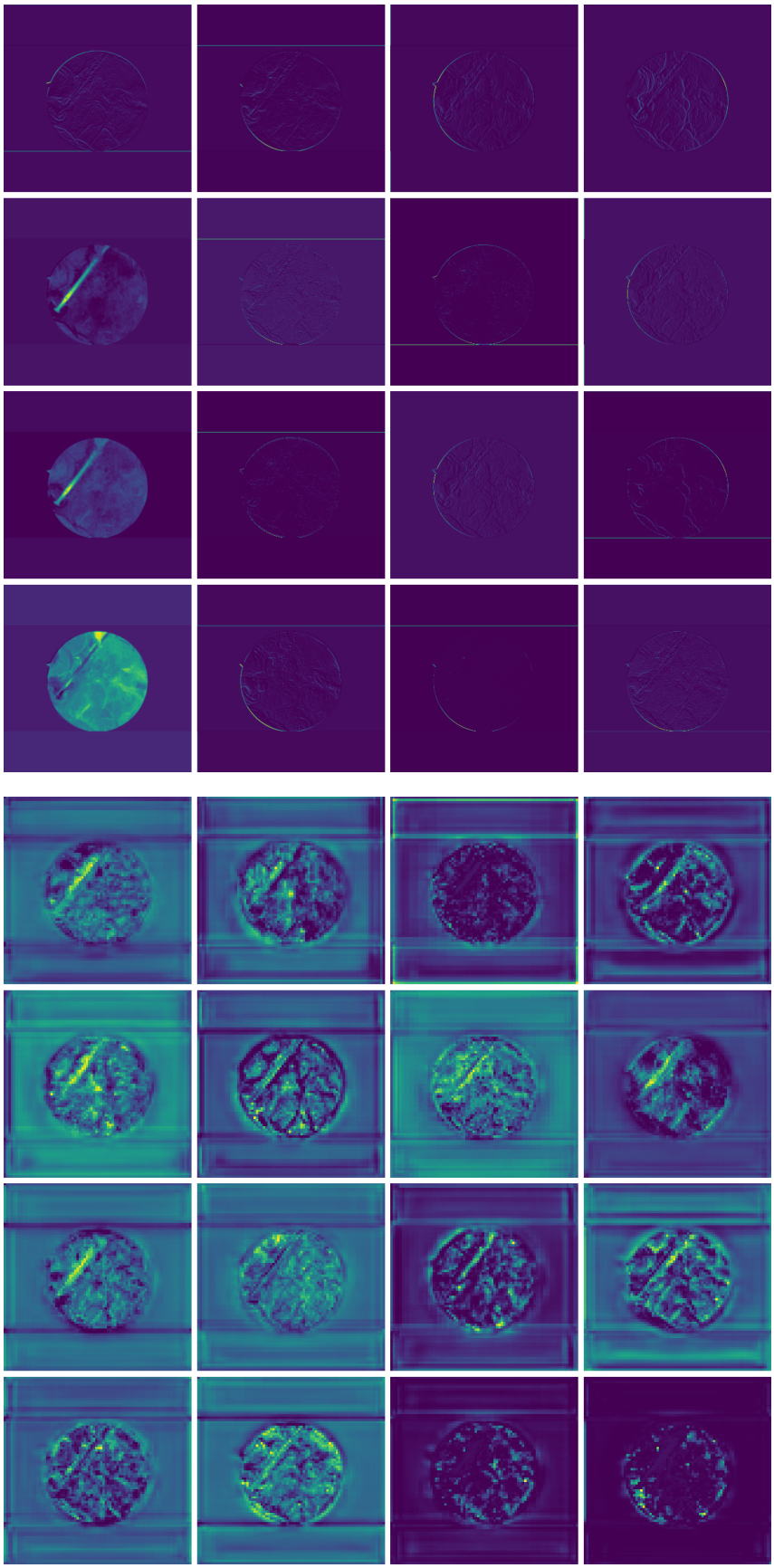

Figure 4.9: Feature-map progression for the same frame. Top: early-stage activations `stage0_Conv`. Bottom: deeper-stage activations `stage16_C3k2`.

The feature-map comparison indicates a depth-dependent transition in representation. Early-layer activations emphasize low-level visual primitives, including local edges and texture transitions. In deeper layers, activations become more structured and selective, with stronger responses over lesion-relevant morphology and comparatively reduced background emphasis. This behavior is consistent with hierarchical feature abstraction in convolutional detectors and supports the observed localization performance.

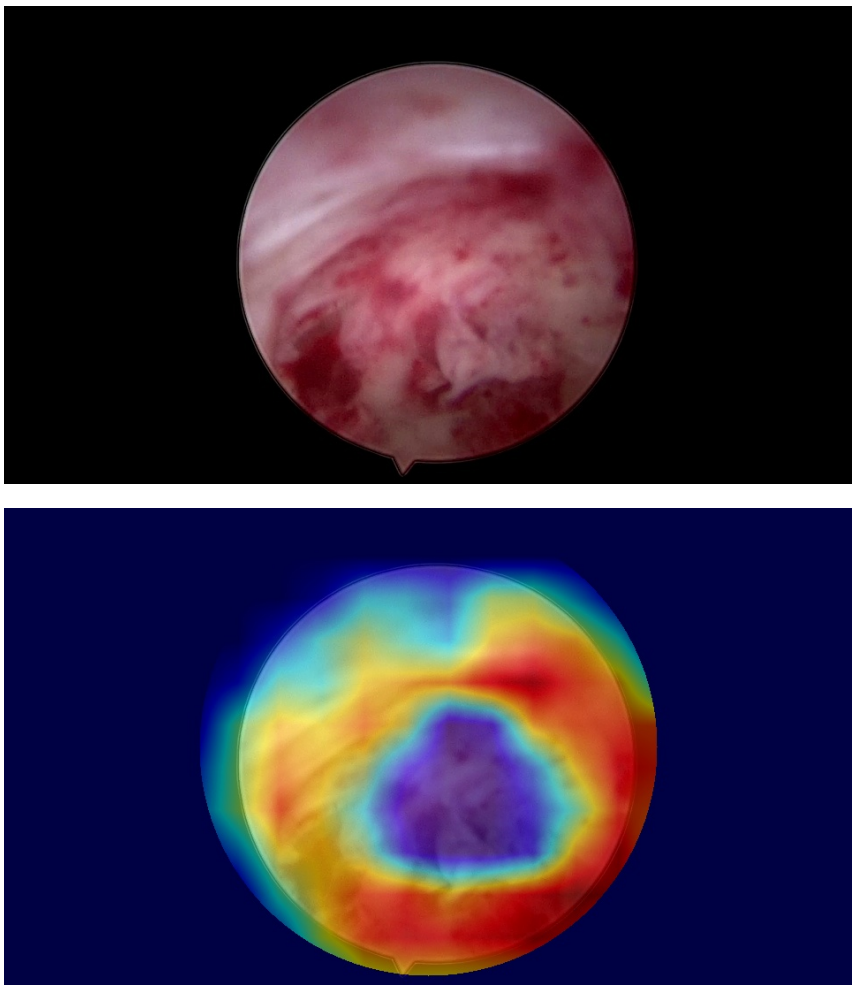

Figure 4.10: Original image (top) and corresponding EigenCAM activation heatmap for a challenging EH frame (bottom).

In this challenging EH frame, activation is distributed over a broader lesion-relevant area rather than being tightly concentrated at a single focal point. The heatmap indicates that model attention follows multiple heterogeneous tissue visual patterns, which is expected in visually complex hysteroscopic scenes with weaker boundary definition and variable local texture. This broader response remains clinically meaningful and is consistent with the increased difficulty observed for challenging lesion categories in the quantitative analysis.

4.3 Hyperparameter Sensitivity Analysis

This section reports controlled comparisons against the baseline in order to quantify sensitivity to key training and model parameters. Each experiment modifies one component at a time while keeping the remaining protocol fixed.

4.3.1 Sensitivity Evaluation Protocol

Table 4.2: Experimental configurations (E0–E9).

Experiment	**Configuration**
E0 Baseline	`model=yolo11m.pt`, `imgsz=640`, `batch=16`, `epochs=200`, `optimizer=AdamW`, `lr0=5e-4`, `cos_lr=True`, `mosaic=0.0`, `mixup=0.0`, `close_mosaic=0`, `scale=0.0`, `cls=1.0`, `box=7.5`, `dfl=1.5`, `EMPTY_TO_POS_RATIO=0.2`.
E1 Input scale	E0 + `imgsz=1280`.
E2 Mosaic augmentation	E0 + `mosaic=1.0`, `close_mosaic=15`.
E4 Strong negative ratio	E0 + `EMPTY_TO_POS_RATIO=0.5`.
E5 Lower learning rate	E0 + `lr0=1e-4`.
E6 Loss reweighting	E0 + `cls=2.0`, `box=6.0`, `dfl=1.0`.
E7 Smaller model	E0 + `model=yolo11s.pt`.
E8 Larger model	E0 + `model=yolo11l.pt`.
E9 Final engineered setting	`model=yolo11m.pt`, `imgsz=640`, `batch=16`, `epochs=200`, `optimizer=AdamW`, `lr0=3e-4`, `cos_lr=True`, `mosaic=0.5`, `mixup=0.0`, `close_mosaic=10`, `scale=0.5`, `cls=1.2`, `box=7.5`, `dfl=1.5`, `EMPTY_TO_POS_RATIO=0.2`.

Since each experiment changes one factor relative to E0, metric differences are attributed to that specific factor under the fixed split protocol.

4.3.2 Quantitative Summary

Table 4.3: Comparative sensitivity analysis results across experiment configurations (mean ± std over folds).

Experiment	test mAP50	test mAP50-95	test Precision	test Recall
E0 Baseline	0.71 ± 0.01	0.40 ± 0.00	0.76 ± 0.03	0.64 ± 0.01
E1 Input Scale	0.52 ± 0.02	0.28 ± 0.01	0.59 ± 0.04	0.49 ± 0.02
E2 Mosaic	0.77 ± 0.01	0.43 ± 0.01	0.78 ± 0.03	0.72 ± 0.02
E4 Neg Ratio 0.5	0.71 ± 0.01	0.40 ± 0.01	0.74 ± 0.04	0.66 ± 0.02
E5 Lower LR	0.77 ± 0.01	0.44 ± 0.00	0.79 ± 0.02	0.71 ± 0.02
E6 Loss Reweight	0.69 ± 0.01	0.38 ± 0.01	0.74 ± 0.05	0.63 ± 0.01
E7 Smaller Model	0.75 ± 0.01	0.42 ± 0.01	0.78 ± 0.01	0.69 ± 0.01
E8 Larger Model	0.68 ± 0.01	0.38 ± 0.01	0.74 ± 0.03	0.61 ± 0.01
E9 Final Engineered	0.81 ± 0.01	0.46 ± 0.01	0.80 ± 0.04	0.76 ± 0.01

4.3.3 Input Resolution Effect (E1)

Increasing input size can, in principle, improve the representation of fine visual structures and support more precise boundary localization in object detection pipelines [23, 28, 29]. In E1, the input resolution was increased relative to E0 while the remaining protocol was kept fixed, to evaluate whether additional spatial detail improves detection quality.

Relative to E0, E1 produced a clear reduction across all principal metrics. On the test split, mAP50 decreased from 0.71 to 0.52, mAP50-95 from 0.40 to 0.28, precision from 0.76 to 0.59, and recall from 0.64 to 0.49.

These results indicate that, under the present dataset scale and training regime, higher resolution did not improve generalization and instead degraded detection quality. Therefore, despite increased computational cost, the E1 setting was not retained in the final engineered configuration.

4.3.4 Mosaic Augmentation Effect (E2)

Data augmentation is used to improve generalization by exposing the model to wider visual variability during training [30, 31]. In YOLO training, mosaic augmentation combines four

different training images into one composite image, which changes object scale and context and can improve robustness to spatial variation. In E2, mosaic was fully enabled (1.0), whereas in E0 it was disabled (0.0). In addition, close-mosaic was activated in E2 (15), meaning mosaic was turned off during the final 15 epochs so the model could finish training on non-mosaic images and better align with the target data distribution. Relative to E0, E2 improved all reported core metrics, with test mAP50 increasing from 0.71 to 0.77, test mAP50-95 from 0.40 to 0.43, test precision from 0.76 to 0.78, and test recall from 0.64 to 0.72. The dominant effect was a strong recall gain, indicating improved sensitivity and better generalization to varied lesion appearance.

4.3.5 Class-Imbalance Effect (E4)

Class imbalance is a central issue in object detection, especially when negative or sparse examples dominate certain training conditions [32–34]. In E4, the empty-label negative sampling ratio was increased to 0.5 relative to E0 (0.2). This change introduces stronger hard-negative exposure during training and is intended to reduce false positive tendency. Relative to E0, E4 produced essentially unchanged test mAP50 (0.71 vs 0.71) and test mAP50-95 (0.40 vs 0.40), while test precision decreased from 0.76 to 0.74 and test recall increased from 0.64 to 0.66. This profile indicates a precision-recall trade-off, with a small sensitivity gain but no clear improvement in aggregate detection accuracy.

4.3.6 Learning-Rate Effect (E5)

The learning rate is a key factor in optimization stability, convergence dynamics, and final generalization performance [24, 25, 35]. In E5, the initial learning rate was reduced to 1e-4 relative to E0 (5e-4), while the optimizer and the remaining training protocol were kept unchanged. This experiment was designed to evaluate whether smaller parameter updates improve convergence quality and downstream detection accuracy. Relative to E0, E5 improved all principal test metrics, with test mAP50 increasing from 0.71 to 0.77, test mAP50-95 from 0.40 to 0.44, test precision from 0.76 to 0.79, and test recall from 0.64 to 0.71. The observed behavior indicates that the lower learning rate provided a more favorable optimization regime in this setting, improving both localization quality and detection reliability without evidence of underfitting.

4.3.7 Loss-Reweighting Effect (E6)

In object detection, training is guided by a composite loss that includes separate terms for class prediction and box localization [6, 36, 37]. Loss reweighting changes how strongly each term influences parameter updates. In practice, increasing the class weight encourages the model to focus more on assigning the correct lesion category, while reducing localization-related weights

lowers the relative emphasis on precise box geometry.

In E6, the balance was shifted toward classification compared with E0. This experiment was designed to test whether stronger class-focused learning improves overall detection behavior. Compared with E0, E6 produced lower values across all principal test metrics: test mAP50 decreased from 0.71 to 0.69, test mAP50-95 decreased from 0.40 to 0.38, test precision decreased from 0.76 to 0.74, and test recall decreased from 0.64 to 0.63. These results indicate that, under the present protocol, reducing the relative localization emphasis weakened detection quality. Therefore, the baseline weighting remains a more suitable balance between classification reliability and localization accuracy.

4.3.8 Smaller Model-Capacity Effect (E7)

Model capacity directly affects the trade-off between detection accuracy and computational efficiency [2, 38, 39]. In E7, the baseline detector was replaced with a smaller model variant (`yolo11s.pt` instead of `yolo11m.pt`), while all remaining settings were kept unchanged relative to E0. This experiment was designed to test whether reduced model capacity can maintain or improve generalization under the same protocol.

Relative to E0, E7 improved all principal test metrics. Test mAP50 increased from 0.71 to 0.75, test mAP50-95 increased from 0.40 to 0.42, test precision increased from 0.76 to 0.78, and test recall increased from 0.64 to 0.69. Although E7 uses a smaller model, this outcome is plausible in limited and imbalanced medical datasets, where lower-capacity models may generalize better by reducing overfitting and providing a more favorable bias-variance trade-off [6, 33, 38, 39]. Under the present experimental conditions, E7 therefore offers a strong accuracy-efficiency profile.

4.3.9 Larger Model-Capacity Effect (E8)

Larger detector variants can capture richer features, but may offer diminishing returns depending on dataset scale and class complexity [2, 39]. In E8, the baseline detector `yolo11m.pt` was replaced with the larger variant `yolo11l.pt` under the same protocol, to test whether increased representational capacity improves detection performance.

Relative to E0, E8 showed lower performance across all principal metrics. Validation mAP50 decreased from 0.71 to 0.65 and validation mAP50-95 from 0.39 to 0.37. On the test split, mAP50 decreased from 0.71 to 0.68, mAP50-95 from 0.40 to 0.38, precision from 0.76 to 0.74, and recall from 0.64 to 0.61.

These results indicate that increasing model capacity did not improve generalization in the present dataset setting. The `yolo11l.pt` configuration introduced higher computational cost

without measurable accuracy gains and was therefore not retained in the final engineered setup.

4.3.10 Engineered Final Configuration (E9)

E9 combines the strongest settings identified during the sensitivity analysis into a single integrated configuration. The final setup uses YOLO11m with image size 640, batch size 16, 200 epochs, AdamW optimizer, and a reduced initial learning rate of 3e-4. It also applies moderate augmentation and loss tuning, including mosaic (0.5), close-mosaic (10), scale (0.5), and class-loss emphasis (cls = 1.2), while preserving the same split protocol used in previous experiments.

Under this configuration, E9 achieved val mAP50 of 0.80 ± 0.02 and val mAP50-95 of 0.46 ± 0.02. On the test split, E9 achieved mAP50 of 0.81 ± 0.01, mAP50-95 of 0.46 ± 0.01, precision of 0.80 ± 0.04, and recall of 0.76 ± 0.01. Relative to the preceding completed experiments, this profile provides the strongest overall balance between detection accuracy, localization quality, and sensitivity. Therefore, E9 is selected as the current best engineered configuration for subsequent reporting and comparison.

4.4 Comparison with RetinaNet-ResNet50

This section compares the selected final engineered YOLO11-based detector with RetinaNet-ResNet50 under the same dataset setting and fold-based evaluation protocol. RetinaNet-ResNet50 is a one-stage anchor-based detector that combines a ResNet-50 backbone, a Feature Pyramid Network (FPN), and task-specific subnetworks for classification and box regression. It also uses focal loss to reduce the dominance of easy negatives and improve learning under class imbalance.

Architecturally, the ResNet-50 backbone extracts hierarchical convolutional features from the input image, progressing from low-level edge/texture patterns to higher-level semantic structures. The FPN then fuses multi-scale backbone features to create a pyramid of semantically strong feature maps, enabling detection of lesions at different sizes. On top of each pyramid level, RetinaNet applies two parallel heads: a classification head that predicts class probabilities for each anchor, and a box-regression head that predicts anchor-to-object coordinate offsets. Final detections are produced after confidence filtering and non-maximum suppression.

To support comparability, both models were trained and evaluated with the same data sources (HS-CMU + HS-CMU-V2), split strategy, and reported metrics (mAP@0.5, mAP@0.5:0.95, precision, recall). An epoch-matched view at 100 epochs was also considered to reduce bias from unequal training duration.

Table 4.4 reports fold-aggregated test results as mean $\pm$ standard deviation under an epoch-

matched setting (100 epochs). Overall, YOLO11m achieved higher detection accuracy and sensitivity than RetinaNet-ResNet50 across all reported metrics.

Table 4.4: Epoch-matched test-set comparison between YOLO11m and RetinaNet-ResNet50.

Model	**test mAP50**	**test mAP50-95**	**test Precision**	**test Recall**
YOLO11m (100 epochs)	0.80 ± 0.01	0.46 ± 0.01	0.78 ± 0.01	0.76 ± 0.00
RetinaNet-ResNet50 (100 epochs)	0.65 ± 0.02	0.33 ± 0.02	0.65 ± 0.02	0.49 ± 0.01

The epoch-matched comparison indicates a clear advantage for YOLO11m. Relative to RetinaNet-ResNet50, YOLO11m improved test mAP50 by 0.15 (0.80 vs 0.65) and test mAP50-95 by 0.13 (0.46 vs 0.33), indicating better overall detection quality and stronger localization performance under stricter IoU criteria. YOLO11m also achieved higher precision (0.78 vs 0.65) and substantially higher recall (0.76 vs 0.49), showing that it both reduces false detections and recovers a larger proportion of true lesions. Under the same 100-epoch training budget, this combined metric pattern indicates a more favorable precision-recall balance and stronger practical sensitivity for hysteroscopic lesion detection.

4.5 Summary of Findings

This chapter presented the experimental results of the proposed hysteroscopic lesion-detection pipeline under a controlled fold-based protocol. Quantitative evaluation across validation and test splits showed that model performance is highly sensitive to training configuration choices, and that improvements are not uniform across all parameter changes.

The sensitivity analysis demonstrated that selected modifications, particularly mosaic-based augmentation and lower learning-rate settings, improved detection quality relative to baseline. In contrast, increasing input resolution and increasing model capacity (`yolo11l.pt`) reduced performance under the present dataset setting, indicating that higher complexity did not translate into better generalization. Loss reweighting toward stronger classification emphasis also degraded aggregate detection results.

The engineered final configuration (E9) achieved the strongest overall balance among completed YOLO experiments, with the highest combined profile in mAP, precision, and recall, and was therefore retained as the primary detector configuration for final reporting.

The comparison subsection further showed that the selected YOLO configuration outperformed the reported RetinaNet-ResNet50 results in the completed evaluations. Overall, these findings

indicate that, for the present data setting, careful configuration and protocol consistency were more decisive than simply increasing model complexity.

5 Discussion

5.1 Introduction

This chapter interprets the experimental findings in relation to the thesis objectives and research questions. It examines what the results imply for model behavior, robustness, and practical applicability in hysteroscopic lesion detection, and situates the observed performance in the context of related studies.

5.2 Interpretation of Main Findings

The sensitivity analysis demonstrated that performance is strongly configuration-dependent and that improvements are not uniform across parameter changes. Certain adjustments improved both mAP and recall, whereas others introduced precision-recall trade-offs or reduced localization quality. This pattern indicates that detector optimization in medical imaging requires controlled evaluation under a fixed and transparent protocol, rather than relying on isolated hyperparameter assumptions.

The engineered final configuration E9 provided the most favorable overall balance among completed experiments, supporting the view that complementary tuning of optimization and augmentation settings can yield cumulative gains. In parallel, class-level behavior remained uneven, with visually ambiguous or less represented categories contributing disproportionately to residual error, indicating that aggregate metrics alone may understate class-specific difficulty.

The model-capacity findings further showed that larger models were not automatically superior in this dataset setting. A smaller variant remained competitive and, in some comparisons, outperformed higher-capacity alternatives. This is consistent with the expectation that, under limited and imbalanced medical data, generalization may improve when model complexity is better matched to available training signal and class distribution.

Finally, comparisons with prior work should be interpreted with protocol awareness. Published studies frequently differ in split strategy, dataset composition, and evaluation pipeline, and therefore external comparisons are best treated as contextual unless strict protocol harmonization is achieved. Within this constraint, the present findings remain methodologically coherent and competitive with recent YOLO-based hysteroscopy detection literature.

5.3 Results Comparison with Other Methods

This section compares the final engineered model with recent hysteroscopy detection methods. The comparison uses commonly reported object-detection metrics (precision, recall, mAP@0.5, and mAP@0.5:0.95) to examine relative strengths and weaknesses in lesion-detection performance. Since these methods are evaluated on hysteroscopic endoscopy imagery, the comparison highlights differences in practical lesion-localization behavior under clinically relevant visual conditions. The external reference methods included are YOLOv11-LC and HL-Det, reported on the HS-CMU dataset [40, 41].

Table 5.1: Contextual comparison with published hysteroscopy detection results.

Method	Precision	Recall	mAP@50	mAP@50-95
YOLO11m E9 final engineered	0.80 ± 0.04	0.76 ± 0.01	0.81 ± 0.01	0.46 ± 0.01
YOLO11m HS-CMU-only (own model)	0.79 ± 0.02	0.76 ± 0.02	0.81 ± 0.02	0.46 ± 0.01
YOLOv11-LC	0.811	0.768	0.815	0.443
HL-Det	0.817	0.755	0.809	0.452

Note: YOLOv11-LC mAP@50-95 is reported as a class-macro estimate computed from published per-class mAP@50-95 values, since a single official aggregate mAP@50-95 value is not provided in the source summary table.

Comparative analysis of results: The contextual comparison indicates that all methods achieve strong precision-recall profiles for hysteroscopic lesion detection. The proposed YOLO11m E9 configuration reports precision of 0.80 ± 0.04 and recall of 0.76 ± 0.01, while the proposed YOLO11m HS-CMU-only variant reports precision of 0.79 ± 0.02 and recall of 0.76 ± 0.02. These values are close to YOLOv11-LC precision 0.811, recall 0.768 and HL-Det precision 0.817, recall 0.755, indicating competitive sensitivity and false-positive control.

For localization quality, both proposed variants achieve mAP@0.5 of approximately 0.81, which is comparable to YOLOv11-LC 0.815 and HL-Det 0.809. Under the stricter mAP@0.5:0.95 criterion, both proposed variants report approximately 0.46, which is higher than HL-Det 0.452

and the reported YOLOv11-LC class-macro estimate 0.443. Overall, the proposed models show a balanced and competitive detection profile across precision, recall, and both mAP criteria.

Since external studies use different split and evaluation protocols, these comparisons are interpreted as contextual evidence rather than strict direct ranking.

Table 5.2: Data-split comparison across related HS-CMU-based studies and the proposed pipeline variants.

Method / Study	**Dataset**	**Split strategy**	**Train**	**Val**	**Test**
HS-CMU dataset baseline (Han et al.) [1]	HS-CMU	Random train/val split	2745	686	N/A
YOLOv11-LC [40]	HS-CMU	Fixed split (7:2:1)	2369	677	339
HL-Det [41]	HS-CMU	Fixed split (7:2:1)	2369	677	339
Proposed YOLO11m pipeline (HS-CMU-only, own model)	HS-CMU	5-fold protocol + inner train/val split	2166	542	677
Proposed YOLO11m pipeline (HS-CMU + HS-CMU-V2, own model)	HS-CMU + HS-CMU-V2	5-fold Stratified-GroupKFold + inner train/val split	3524.6 ± 7.0	927	1158

Explanation: For the HS-CMU dataset baseline [1], a random train/validation split is reported (2745/686), while an independent test split is not explicitly specified in the same setup. Both YOLOv11-LC [40] and HL-Det [41] report a fixed 7:2:1 split on HS-CMU (2369/677/339). The proposed pipeline is reported in two internal variants developed in this dissertation: an HS-CMU-only configuration and an HS-CMU+HS-CMU-V2 configuration. For both variants, values represent effective per-fold usage as mean $\pm$ standard deviation across five folds under a fixed protocol and seed.

5.4 Limitations and Practical Implications

Several limitations should be considered when interpreting the reported results. Although fold-based evaluation was applied, the study remains constrained by dataset scale, class imbalance, and class-level heterogeneity, with lower-representation categories showing less stable behavior. The experiments were conducted on HS-CMU and HS-CMU-V2 under a fixed internal protocol, so external generalization to different hospitals, devices, operators, and imaging conditions

was not directly validated. In addition, the analysis is frame-based and detector-focused, and does not include full temporal clinical workflow validation, inter-observer variability analysis, or prospective real-time deployment assessment. As also reflected in recent hysteroscopy literature, differences in split strategy, annotation policy, and evaluation setup limit strict cross-study comparability, especially when reported metrics are not fully aligned.

Further limitations arise from the imaging domain itself. Compared with natural-image detection benchmarks, hysteroscopic lesions often present weak or gradual boundaries, low lesion-to-background contrast, and specular highlights that can obscure informative structures. Lesions also exhibit substantial intra-class and inter-class variability, including small and subtle early-stage abnormalities, which increases representation difficulty and can reduce robustness. In this context, standard multi-scale fusion mechanisms may not always capture cross-scale dependencies strongly enough to optimize localization precision and class discrimination simultaneously in difficult cases.

Practical constraints were also significant. Building a new clinically validated dataset is highly time-consuming and requires sustained clinician-expert involvement for annotation, quality control, and consensus verification. Large-scale training and repeated ablation cycles are computationally expensive, with long runtime requirements that can limit the number of experiments completed within project timelines.

Despite these limitations, the findings have practical significance. The selected YOLO11m configuration achieved a strong precision-recall balance and robust localization quality, supporting its role as a computer-aided decision-support component for hysteroscopic review. In practical use, such a system can improve detection consistency and reduce missed focal findings, while remaining an assistive tool that complements, rather than replaces, clinician judgment.

5.5 Summary

This chapter interpreted the experimental findings in relation to the thesis objectives and research questions. The analysis showed that detection performance is strongly configuration-dependent and that the engineered YOLO11m setting offers a favorable balance of precision, recall, and localization quality. Comparative analysis indicated that the proposed YOLO11m variants developed in this dissertation, including the HS-CMU+HS-CMU-V2 and HS-CMU-only configurations, remain competitive with recent hysteroscopy detectors such as YOLOv11-LC and HL-Det. At the same time, interpretation was made with protocol awareness, since differences in split design, dataset composition, and evaluation setup limit strict direct cross-study ranking. Limitations related to dataset scale, class imbalance, domain-specific visual difficulty, and protocol dependence were identified. These findings establish the basis for the concluding remarks

and future research directions presented in Chapter 6.

6 Conclusions and Future Work

6.1 Concluding Remarks

This thesis addressed computer-aided detection of endometrial lesions in hysteroscopic imagery through a reproducible deep-learning pipeline based on YOLO11. The central objective was to design and evaluate a detection framework that is methodologically consistent, quantitatively transparent, and practically relevant for lesion localization support.

An end-to-end workflow was implemented, including dataset integration from HS-CMU and HS-CMU-V2, annotation consistency checks, fold-based data partitioning, model training, validation, test evaluation, and qualitative visualization. All experiments were conducted under a fixed protocol with fold-aggregated reporting to ensure comparability across configurations.

The results showed that detector performance was highly sensitive to training configuration choices. Controlled ablations demonstrated that changes in input resolution, augmentation policy, optimization settings, class-imbalance handling, loss weighting, and model capacity did not produce uniform effects. Some settings improved sensitivity and localization quality, whereas others degraded precision-recall balance despite increased computational cost. These findings indicate that robust medical detection performance should be established through protocol-controlled evidence rather than assumed from default settings or general benchmark trends.

The engineered final configuration E9 achieved the strongest overall performance profile among completed experiments and provided the best balance across precision, recall, mAP@0.5, and mAP@0.5:0.95. Comparative analysis also showed a clear advantage over the epoch-matched RetinaNet-ResNet50 baseline under the same internal protocol. Contextual comparison with recent hysteroscopy-focused detectors further indicated that the proposed configuration remains competitive within current literature-level performance ranges.

Beyond metric outcomes, this work contributes a methodological framework for reliable detector assessment in limited and imbalanced medical datasets. The distinction between internally matched comparisons and contextual external references is particularly important for scientific

validity, given that differences in split strategy, annotation policy, and reporting conventions can substantially affect apparent ranking across studies.

Class-level behavior also provided important insight. Although aggregate metrics were strong, performance was not uniform across lesion categories, and difficult classes remained associated with lower stability and higher confusion. This supports the need for class-aware optimization and data-centric refinement in subsequent development phases.

A clinically important next step is a dedicated endometritis-focused extension, since the present work optimized multiclass lesion detection but did not include an endometritis-specific training protocol with frame-level ground truth.

From a practical standpoint, the developed pipeline demonstrates that YOLO-based detectors can provide strong lesion-detection performance in hysteroscopic imagery while maintaining a computational profile compatible with future near-real-time assistance settings. At the same time, the intended role remains assistive, supporting clinician review and consistency rather than replacing specialist judgment.

6.2 Future Work

Several directions can extend the present work and strengthen both methodological rigor and clinical relevance.

A complete consolidation of pending experiment tracks under identical protocol conditions would improve confidence in ranking stability and reduce uncertainty from partially explored settings. In parallel, a strict direct-comparability benchmark track aligned with selected prior studies at the split and reporting level would support clearer literature-level interpretation.

Architecture-level extensions remain a natural continuation, particularly attention-enhanced and multi-scale refinement variants evaluated under the same protocol discipline established in this thesis. Such studies should preserve balanced reporting across accuracy, parameter count, FLOPs, and latency, so that practical deployment implications remain visible alongside metric gains.

Further progress can also be achieved through structured interpretability and error analysis. Systematic inspection of false positives and false negatives, together with EigenCAM and feature-map visualization, can clarify whether failure modes are dominated by weak boundaries, low contrast, specular highlights, subtle early-stage patterns, or annotation uncertainty. This type of evidence can guide targeted corrective interventions more effectively than broad hyperparameter search.

Data-centric refinement is equally important. Expansion of under-represented classes, multi-reader annotation reconciliation, hard-case mining, and stronger label quality control are likely

to improve class-level robustness and reduce variance across folds. Because clinically validated annotation is time-intensive and requires sustained expert involvement, practical data strategy should be treated as a core component of model improvement rather than a secondary step.

Given the inherently video-based nature of hysteroscopy, temporal modeling represents another high-impact extension. Sequence-aware inference and temporal consistency constraints may improve prediction stability compared with independent frame-level detection and may reduce isolated false alarms in difficult scenes.

A priority continuation is a dedicated endometritis-focused, video-centric extension. A practical pilot direction is to leverage available clinical videos using temporally structured segment-level learning, with progressive refinement through targeted expert review where needed. Given the limited availability of dense frame-level annotations, weakly supervised and hybrid annotation strategies should be investigated to support reliable model development while maintaining clinical realism. To strengthen discrimination and reduce bias, future datasets should include both endometritis-positive material and clinically relevant negative controls. Additional improvement may come from complementary texture-aware modeling and carefully controlled synthetic augmentation, evaluated as supportive components alongside real clinical data.

It may also be possible to investigate CPU-oriented deployment by evaluating lightweight YOLO variants, model compression, quantization, and optimized inference backends, with the aim of improving runtime feasibility in settings where dedicated GPUs are not available.

Clinical translation requires dedicated prospective evaluation. Deployment-oriented studies should examine threshold calibration policies, interface design for visual overlays, runtime behavior under workflow constraints, and clinician interaction patterns. Pilot reader-assistance studies can then assess whether algorithmic gains translate into improved detection consistency and review efficiency in practice.

A longer-term direction is personalized AI-assisted hysteroscopy through multimodal integration of visual findings with patient-level metadata, histopathology, and clinical history, to support more individualized risk-aware decision support.

6.3 Closing Statement

This thesis developed and evaluated a reproducible deep-learning framework for endometrial lesion detection in hysteroscopic imagery and provided a structured sensitivity analysis across key training and model parameters. The findings show that careful configuration design, protocol consistency, and fold-aggregated evaluation can produce strong and reliable detection performance in a challenging medical domain.

At the same time, the study identifies clear technical and practical boundaries related to dataset

scale, class imbalance, visual ambiguity, annotation burden, and validation scope. These boundaries define a concrete roadmap for continued development.

With sustained data-centric refinement, controlled architecture evolution, and prospective clinical validation, the presented framework can support progression toward robust and clinically useful computer-aided hysteroscopic decision-support systems. A key next step is an endometritis focused video-centric extension, combining temporally structured learning, progressive annotation refinement, and clinically grounded evaluation.

BIBLIOGRAPHY

[1] R. Han, Y. Xie, K. You, L. Cao, and H. Li, "The hs-cmu dataset for diagnosing benign and malignant diseases through hysteroscopy," *arXiv preprint arXiv:2406.02908*, 2024. [Online]. Available: https://arxiv.org/abs/2406.02908

[2] Ultralytics, "Yolo11 documentation and official benchmarks," https://docs.ultralytics.com/models/yolo11/, 2026, accessed: 2026-04-21.

[3] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *CoRR*, vol. abs/2410.17725, 2024. [Online]. Available: https://arxiv.org/abs/2410.17725

[4] Ultralytics, "Ultralytics yolo documentation," https://docs.ultralytics.com/, 2026.

[5] G. Garcia, S. Lauria, P. Sanchez, and H. Fujita, "Layoutyolo: A new deep learning-based document layout analysis pipeline for small datasets," *Applied Sciences*, vol. 15, no. 6, p. 3164, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/15/6/3164

[6] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023. [Online]. Available: https://doi.org/10.1109/JPROC.2023.3238524

[7] Y. Zhang, Z. Wang, J. Zhang, C. Wang, Y. Wang, H. Chen, L. Shan, J. Huo, J. Gu, and X. Ma, "Deep learning model for classifying endometrial lesions," *Journal of Translational Medicine*, vol. 19, no. 1, p. 10, 2021. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7788977

[8] Y. Takahashi, K. Sone, K. Noda, K. Yoshida, Y. Toyohara, K. Kato, F. Inoue, A. Kukita, A. Taguchi, H. Nishida, M. Maruyama, Y. Osuga, and T. Fujii, "Automated system for diagnosing endometrial cancer by adopting deep-learning technology in hysteroscopy," *PLOS ONE*, vol. 16, no. 3, p. e0248526, 2021. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0248526

[9] A. Zhao, X. Du, S. Yuan, W. Shen, X. Zhu, and W. Wang, "Automated detection of endometrial polyps from hysteroscopic videos using deep learning," *Diagnostics*, vol. 13, no. 8, p. 1409, 2023. [Online]. Available: https://www.mdpi.com/2075-4418/13/8/1409

[10] M. Mascarenhas, C. Peixoto, R. Freire, J. Cavaco Gomes, P. Cardoso, I. Castro, M. Martins, F. Mendes, J. Mota, M. J. Almeida *et al.*, "Artificial intelligence and hysteroscopy: A multicentric study on automated classification of pleomorphic lesions," *Cancers*, vol. 17, no. 15, p. 2559, 2025. [Online]. Available: https://www.mdpi.com/2072-6694/17/15/2559

[11] M. G. Munro, H. O. D. Critchley, M. S. Broder, I. S. Fraser, and F. W. G. on Menstrual Disorders, “Figo classification system (palm-coein) for causes of abnormal uterine bleeding in nongravid women of reproductive age,” *International Journal of Gynecology & Obstetrics*, vol. 113, no. 1, pp. 3–13, 2011. [Online]. Available: https://doi.org/10.1016/j.ijgo.2010.11.011

[12] Jiyuan Medical and Beijing Chaoyang Hospital, Capital Medical University, “Hs-cmu dataset release page (openxlab, including update records),” https://openxlab.org.cn/datasets/jiyuanmedical/HS-CMU/tree/main, 2025, accessed: 2026-04-15.

[13] G. Jocher and J. Qiu, “Ultralytics yolo11,” 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[14] P. Burai, A. Hajdu, E. M. Felipe-Riveron, and B. Harangi, “Segmentation of the uterine wall by an ensemble of fully convolutional neural networks,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 49–52. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/30440338/

[15] B. Li, H. Chen, and H. Duan, “Visualized hysteroscopic artificial intelligence fertility assessment system for endometrial injury: an image-deep-learning study,” *Annals of Medicine*, p. 2478473, 2025. [Online]. Available: https://doi.org/10.1080/07853890.2025.2478473

[16] P. Török and B. Harangi, “Digital image analysis with fully connected convolutional neural network to facilitate hysteroscopic fibroid resection,” *Gynecologic and Obstetric Investigation*, vol. 83, no. 6, pp. 615–619, 2018. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29975937/

[17] L.-h. He, Y.-z. Zhou, L. Liu, W. Cao, and J.-h. Ma, “Research on object detection and recognition in remote sensing images based on yolov11,” *Scientific Reports*, 2024. [Online]. Available: https://doi.org/10.1038/s41598-025-96314-x

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/papers/Kirillov_Segment_Anything_ICCV_2023_paper.pdf

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91

[20] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-guided feature aggregation for video object detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 408–417. [Online]. Available: https://arxiv.org/abs/1703.10025

[21] Y. Chen, Y. Cao, H. Hu, and L. Wang, “Memory enhanced global-local aggregation for video object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 337–10 346. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Chen_Memory_Enhanced_Global-Local_Aggregation_for_Video_Object_Detection_CVPR_2020_paper.html

[22] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “Cspnet: A new backbone that can enhance learning capability of cnn,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1571–1580. [Online]. Available: https://doi.org/10.1109/CVPRW50498.2020.00203

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015. [Online]. Available: https://doi.org/10.1109/TPAMI.2015.2389824

[24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[25] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755. [Online]. Available: https://doi.org/10.1007/978-3-319-10602-1_48

[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *CVPR*, 2017, pp. 2117–2125.

[29] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *ICML*, 2019.

[30] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *ICCV*, 2017, pp. 2980–2988.

[33] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: A review,” *TPAMI*, vol. 43, no. 10, pp. 3388–3415, 2020.

[34] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in *CVPR*, 2016, pp. 761–769.

[35] L. N. Smith, “Cyclical learning rates for training neural networks,” in *WACV*, 2017, pp. 464–472.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in *CVPR*, 2019, pp. 658–666.

[37] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-iou loss: Faster and better learning for bounding box regression,” *AAAI*, vol. 34, no. 07, pp. 12 993–13 000, 2020.

[38] A. Howard, M. Sandler, G. Chu, and et al., “Searching for mobilenetv3,” in *ICCV*, 2019, pp. 1314–1324.

[39] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *CVPR*, 2020, pp. 10 781–10 790.

[40] J. Zhang and X. L. Peng, “Yolov11-lc: accurate detection of benign and malignant endometrial lesions using lca-enhanced yolov11 with cma-c3k2,” *Signal, Image and Video Processing*, vol. 20, p. 15, 2026, published online 14 January 2026; includes HS-CMU comparison and per-class detection results.

[41] X. Chen, G. Wang, J. Cao, and Y. Lu, “Hl-det: A multi-scale adaptive network integrating spatial-frequency features for endometrial lesion detection,” *Biomedical Signal Processing and Control*, 2025, includes HS-CMU comparison and per-class detection results.

APPENDICES

APPENDIX A

Reproducibility Artifacts

This appendix provides implementation-level evidence for the experimental protocol and computing environment reported in Chapter 4.

A.1 Training Pipeline Configuration

The following excerpts summarize the core protocol settings used in the Python training pipeline.

```
NUM_RUNS = 1
NUM_FOLDS = 5
RANDOM_SEED = 42

USE_EMPTY_NEG_SAMPLING = True
EMPTY_TO_POS_RATIO = 0.2

YOLO_MODEL = "yolo11m.pt"
EPOCHS = 200
OPTIMIZER = "AdamW"

sgkf = StratifiedGroupKFold(NUM_FOLDS, shuffle=True, random_state=run)

tr_vids, va_vids, tr_lbls, va_lbls, tr_ids, va_ids = train_test_split(
    train_vids, train_lbls, train_ids, test_size=0.2, random_state=RANDOM_SEED
)

if val_map50 > global_best_val_map50:
    global_best_val_map50 = val_map50
    global_best_model_path = best_model_path
```

A.2 SLURM Job Configuration

The following excerpts summarize the batch-resource requests and runtime initialization used in the SLURM scripts.

```
#SBATCH --partition=GPU
#SBATCH --gres=gpu:v100:1
#SBATCH --cpus-per-task=8
#SBATCH --mem=40G
#SBATCH --time=30:00:00

module purge
module load Python/3.10.8-GCCcore-12.2.0
module load CUDA/11.7.0

VENV_PATH=$HOME/venvs/yolo11_env
source $VENV_PATH/bin/activate

export PYTORCH_CUDA_ALLOC_CONF="max_split_size_mb:512"
```

A.3 Runtime Environment Snapshot

The following system and software snapshot was collected from the same environment used for experiment execution.

```
Host: turing1.ucy.ac.cy

CPU: Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz
CPU(s): 32
Socket(s): 2
Core(s) per socket: 16
Thread(s) per core: 1

Memory (RAM): 93G total

GPU (from training logs):
Tesla V100-SXM2-32GB, 34.1 GB

Python: 3.10.8
PyTorch: 2.0.1+cu117
Ultralytics: 8.4.9
CUDA toolkit: 11.7
```

```
SchedulerType: sched/backfill
SelectType: select/cons_res
SelectTypeParameters: CR_CORE_MEMORY
```