Evaluating the Applicability of Real-Time Object Detection Algorithms on Video Recordings of Clinical Colonoscopy Procedures

Master Thesis

Master of Science in Applied Computer Science

Andreas Franz Schwab

January 2, 2026

Supervisor:

1st: Prof. Dr. Christian Ledig
2nd: M.Sc Sebastian Dörrich

Chair of Explainable Machine Learning
Faculty of Information Systems and Applied Computer Sciences
Otto-Friedrich-University Bamberg

Abstract

Object detection is widely used in computer vision and has important applications in computer-aided diagnosis for colonoscopy, where missed lesions are clinically relevant and strongly linked to cancer risk. Recent work has addressed this problem by applying real-time object detection models to curated image datasets or short video segments. Unfortunately, these evaluation settings simplify the task and provide limited insight into detector behavior on full-length clinical procedures, where negative frames dominate and visual conditions vary substantially. Moreover, reported results are often difficult to compare due to missing information on training strategies, dataset splits, operating thresholds, or the use of non-public training data.

In contrast, this thesis assesses the applicability of selected object detection algorithms on video recordings of clinical colonoscopy. Using the REAL-Colon dataset, four representative architectures, Faster R-CNN, YOLOv8, YOLOv11, and RT-DETR, are evaluated under explicitly defined conditions. This provides a basis for studying detector behavior under realistic procedural constraints and shows performance characteristics that are masked in simplified benchmarks. The analysis reveals that, although competitive performance is often reported on curated datasets, detector effectiveness decreases markedly on complete colonoscopy videos. A direct architectural comparison identifies a clear trade-off: Transformer-based models (RT-DETR) demonstrate superior temporal consistency and sensitivity, whereas CNN-based architectures (YOLOv11) offer higher throughput and specificity. However, all evaluated architectures have limitations in stable early detection, struggling to identify lesions during the initial seconds of appearance. Regarding data composition, the experiments demonstrate that models trained on full-procedure video learn robust features that transfer effectively to external datasets, whereas models trained on curated clips fail to generalize to the variability of full procedures. The results provide a realistic and reproducible assessment of the current capabilities and limitations of real-time polyp detection systems in clinical colonoscopy and offer a clear basis for future work targeting clinically applicable detection methods.

Acknowledgements

Special thanks are given to all patients whose anonymized colonoscopy data was made available for research across the datasets used in this thesis. I want to particularly acknowledge the patients and the Basque Biobank for providing the PICCOLO dataset (www.biobancovasco.bioef.eus). Thanks also to Frederik for constant support and helping maintain perspective and sanity throughout the thesis writing process.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
SOTA	State-of-the-art
CAD	Computer Aided Detection
CRC	Colorectal Cancer
HP	Hyperplastic Polyp
AD	Adenoma
SSP	Sessile Serrated Polyp
TSA	Traditional Serrated Adenoma
ADR	Adenoma Detection Rate
SVM	Support Vector Machine
Bbox	Bounding Box
GT	Ground-truth
HOG	Histogram of Gradients
DPM	Deformable Part-based Model
CNN	Convolutional Neural Network
R-CNN	Regional-Based CNN
VOC	Visual Object Classes
COCO	Common Objects in Context
SDD	Single Shot MultiBox Detector
YOLO	You Only Look Once
DETR	Detection Transformer
RT-DETR	Real-Time Detection Transformer
NMS	Non-Maximum Suppression
SPP	Spatial Pyramid Pooling
SPPF	Faster Spatial Pyramid Pooling
RPN	Region Proposal Network
FPN	Feature Pyramid Pooling
PANet	Path Aggregation Network
IoU	Intersection-over-Unit
GIoU	Generalized Intersection-over-Unit
DIoU	Distance Intersection-over-Unit
CIoU	Complete Intersection-over-Unit
NAS	Neural Architecture Search
CSPnet	Cross Stage Partial Network
AIFI	Attention-Based Intra-Scale Feature Interaction
CCFF	ross-Scale Feature Fusion
ROC	Receiver Operating Characteristic
FROC	Free-response ROC
AFROC	Alternative FROC

1 Introduction

Colonoscopy is the most effective tool for the early detection and prevention of colorectal cancer (CRC), which remains one of the leading causes of cancer-related mortality worldwide (Morgan et al., 2023). The procedure relies on direct visual inspection of the colonic mucosa and allows immediate removal of precancerous lesions during the examination (Bretthauer et al., 2022; Münzer et al., 2018). The diagnostic outcome, however, depends heavily on human perception during a visually demanding, real-time procedure. Even under optimal conditions, lesions may be missed due to subtle appearance, short visibility intervals, or momentary distraction (Wang et al., 2019). These limitations have motivated computer-aided detection systems that aim to support endoscopists by automatically highlighting suspicious regions or frames during the examination.

In recent years, deep learning–based object detectors have been increasingly applied to automated polyp detection, achieving high accuracy while operating at frame rates compatible with real-time colonoscopy (Misawa et al., 2021). Many of these systems are evaluated on curated datasets with still images or short video clips, where lesions are frequent and visual conditions are comparatively controlled. Full-length colonoscopy procedures differ from these settings. The majority of frames contain no lesions, and detector outputs occur in long sequences that have strong variation in appearance and motion. Under these conditions, detector behavior is determined not only by localization accuracy, but also by false alerts on negative frames and the temporal consistency of detections when lesions are visible (Biffi et al., 2024). Systematic evaluations that capture these aspects on complete colonoscopy videos remain limited, and comparisons across studies are often complicated by insufficient reporting of dataset composition, training protocols, and operating thresholds. As a result, it remains unclear how performance gains reported on curated benchmarks work in realistic clinical use.

This thesis addresses this gap by evaluating the applicability of teal-time object detection algorithms on video recordings of clinical colonoscopy procedures. The primary focus lies on the REAL-Colon dataset, which contains full-length examinations annotated at both frame and lesion level (Biffi et al., 2024). Additionally, the image- and clip-based datasets PICCOLO (Sánchez-Peralta et al., 2020) and SUN (Misawa et al., 2021) are used as reference points to contrast detector behavior across levels of dataset complexity. To capture the model performance, the experimental setup integrates metrics at the detection-, frame-, and lesion-level. This layered approach assesses not only technical localization accuracy but also clinically relevant factors such as temporal stability, latency to first detection, and false-positive behavior over time. Beyond establishing a transparent and reproducible baseline with unmodified detector configurations, this work examines the generalization behavior of detectors trained on continuous procedural video versus curated clips or images. The analysis investigates potential asymmetries in domain transfer, with focus on the role of full-procedure video data for robust feature learning. Furthermore, it characterizes the trade-offs between modern Transformer-based and CNN-based architectures regarding temporal consistency, latency, and resource efficiency, providing a reference on which future work can assess architectural adaptations under procedure-level conditions.

The thesis is structured as follows: Chapter 2 introduces the medical background of colorectal screening and lesion detection. Chapter 3 provides an overview of object detection methods and evaluation concepts relevant to this work. Related datasets and prior studies are discussed in Chapter 4, followed by a detailed description of the datasets used in Chapter 5. The experimental methodology is outlined in Chapter 6, and results are presented in Chapter 7. The findings are dicsussed in Chapter 8, followed by the limitations in Chapter 9, and conclusions in Chapter 10.

2 Medical Background

This chapter introduces the medical foundations necessary to understand the clinical relevance of colonoscopy and automated polyp detection. It outlines key concepts related to colorectal lesions, screening procedures, and diagnostic performance measures and summarizes the clinical factors that motivate the use of computer-aided detection systems as supportive tools during colonoscopic examinations.

2.1 Colorectal Lesions and Polyps

In colonoscopy, the term *lesion* is used broadly for any visible abnormality of the colonic mucosa. Clinically, these are first distinguished by their biological behaviour. *Non-neoplastic* lesions, most commonly hyperplastic polyps (HP), carry little to no potential for malignant transformation (Rex et al., 2017), whereas *neoplastic* lesions, in contrast, are precursors that can progress to colorectal cancer if left untreated. Following Rex et al. (2017), neoplastic lesions fall into two major families: *conventional adenomas* and *serrated neoplastic lesions*. Conventional adenomas (AD) are characterized by dysplasia and represent the established precursor pathway to colorectal cancer (Bernal et al., 2017; Rex et al., 2017). The serrated spectrum is more complex: while hyperplastic polyps are generally benign, sessile serrated polyps (SSP) and traditional serrated adenomas (TSA) are considered precancerous lesions (Rex et al., 2017). Unlike adenomas, SSPs may lack cytologic dysplasia yet still carry significant malignant potential (Rex et al., 2017).

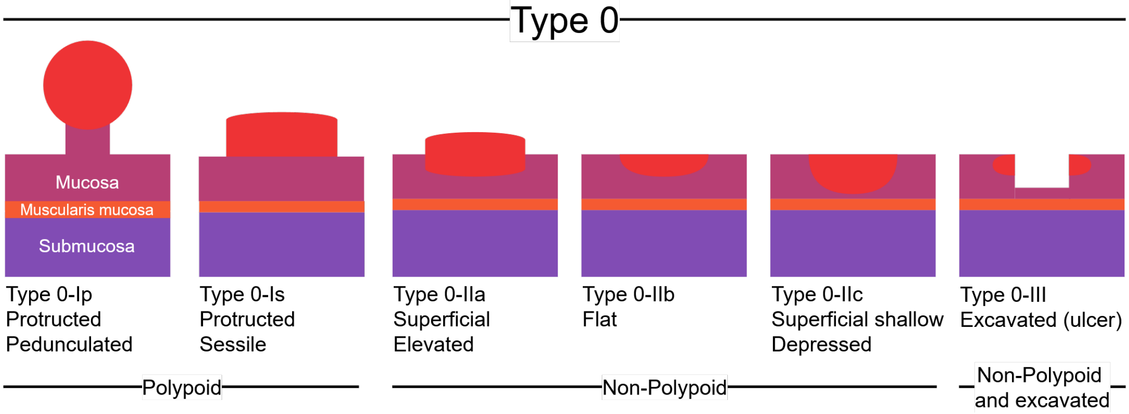


Figure 1: Schematic illustration of the Type 0 superficial lesions of the colon.

Participants in the Paris Workshop (2003) provide a complementary morphological description for Type-0 lesions. They define *superficial neoplastic lesions* as those confined to the mucosa or submucosa, a stage associated with a low risk of lymphatic spread. These are grouped into *polypoid* (Type 0–I), *non-polypoid* (Type 0–II) or *non-polypoid and excavated* (Type 0–III) morphologies. Polypoid lesions protrude above the mucosa and include pedunculated (0-Ip) and sessile (0-Is) forms. Non-polypoid lesions show little or no protrusion and may appear slightly elevated (0-IIa), completely flat (0-IIb), depressed (0-IIc) or excavated (0-III), with depressed lesions carrying a particularly high malignant potential. Figure 1 illustrates these variants. While mixed subtypes do occur, this overview focuses on the primary classifications only.

For the purpose of this thesis, the terms *lesion* and *polyp* are used interchangeably to denote a Region of Interest (RoI). Conceptually, every polyp is a lesion, but not every lesion forms a protruding polypoid structure. The primary objective is the detection of all mucosal anomalies, from protruding or flat to neoplastic or non-neoplastic, to ensure no potential precursor is missed by the endoscopist.

2.2 Fundamentals of Colorectal Screening

Morgan et al. (2023) identify colorectal cancer (CRC) as a major global health concern, ranking as the third most common cancer and the second leading cause of cancer-related deaths in 2020. Most cases follow the adenoma–carcinoma sequence, in which benign polyps slowly develop into malignancy over many years, providing a critical window for early identification and elimination Rex et al. (2017).

Fortunately, there exist prevention techniques, such as screening (Morgan et al., 2023). These approaches can be grouped by stool-based tests, such as the guaiac-based fecal occult blood tests (gFOBTs) or the fecal immunochemical test (FIT) and direct visualization tests (Ladabaum et al., 2020). Stool tests are non-invasive and scalable, yet colonoscopy remains the gold standard as it allows real-time inspection of the mucosa and immediate removal of neoplastic tissue (Bretthauer et al., 2022; Münzer et al., 2018). During the procedure, a flexible endoscope is guided through the colon, enabling the physician to detect and resect lesions in real-time, contributing to the reduction of CRC incidence and mortality (Münzer et al., 2018). The effectiveness of colonoscopy has already been verified about four decades ago with a trial by (Winawer et al., 1993). For their study they invited 1418 patients with "at least one histologically documented adenoma" (Winawer et al., 1993), removed all detected adenomas, and followed them for three to six years. A comparison with a control group in which adenomas were not resected would not have been ethically permissible, so the authors compared the observed cancer incidence with several reference populations instead (Winawer et al., 1993). During follow-up inspections, only five CRCs occurred, corresponding to a roughly 70–90% lower incidence than expected from these reference groups, demonstrating the preventive effect of complete resection. The more recently large-scale randomized NordICC study (Bretthauer et al., 2022) included approximately 84,000 participants who were divided into two groups. One group was invited to undergo a single preventive colonoscopy. After 10 years, it was observed that participants in the selected group who actually did the procedure had a 31% reduction in the incidence of CRC and a roughly 50% reduction in CRC-related mortality (Bretthauer et al., 2022).

Despite these benefits, colonoscopy is not perfect. Its result depends on human perception, and several factors can lead to missed lesions. Wang et al. (2019) describe hypotheses for this cause such as variation in operator skill and cognitive effects like inattentional or change blindness, where relevant findings are overlooked due to distraction or brief disruptions in visual scanning. Technical limitations further complicate detection. As noted by Münzer et al. (2018), reflections from wet mucosa, uneven illumination, specular highlights, geometric distortion, and transient tissue deformation can alter local contrast or obscure detail. Motion blur is also common with up to a quarter of frames being diagnostically limited due to rapid camera movement. Some common challenging examples are shown in Figure 2 These issues are reflected in example colonoscopy studies, where polyp miss rates of up to 27% have been reported (Mahmud et al., 2015; Zhao et al., 2019a). These missed lesions are clinically significant, making them a valuable metric for colonoscopy. There is a strong correlation between the Adenoma Detection Rate (ADR), so the proportion of screening colonoscopies where at least one adenoma is identified, and the risk of interval cancer (cancer diagnosed between scheduled screenings). “With each 1.0% increase in adenoma detection rate (ADR), there is an associated 3.0% decrease in the risk of interval CRC”(Wang et al., 2019; Corley et al., 2014).

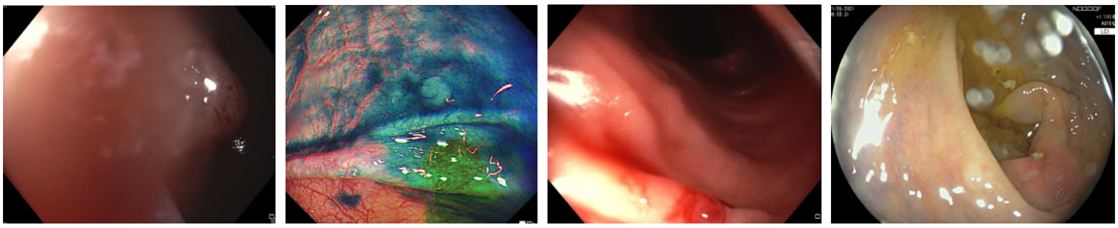


Figure 2: Representative colonoscopy frames from the REAL-Colon dataset with common visual challenges.

2.3 History of Computer Aided Polyp Detection

The above mentioned limitations have motivated the development of systems that should act as a real-time “second observer” (Wang et al., 2019), helping to classify potential cancerous polyps and proposing potential RoIs through bounding boxes or segmentation masks.

Early approaches rely on integrating hand-crafted features designed to match simple structure, color or texture of a potential polyp, and classical machine learning classifiers. Kang and Doraiswami (2003) boost image quality prior to classification. The first step is contrast enhancement by applying edge detection on R,G and B channels of the image separately resulting in a thick white outline shape for any potential polyp. The outputs are then classified to either be a polyp or not by comparing the features to a set of expected characteristics, such as area, original image color and elliptical shape Kang and Doraiswami (2003). Hwang et al. (2007) build on the insight of polyps having an elliptical shape with high or low edge information. The authors first group promising edges and segment the image into multiple regions. Those regions are then formed to an ellipse using the edge infomration. In the last step only ellipses that resemble a polyp are kept. The criteria spans curve direction, edge distance and intensity value Hwang et al. (2007). Ameling et al. (2009) compare multiple texture-based feature extractors. Namely grey-level co-occurrence matrices (GLCMs), which "describe[s] how often different combinations of pixel values occur in an image" Ameling et al. (2009) and local binary patterns (LBP), which threshold each 3×3 neighborhood against the center pixel and creates a single value for this region. As a classification technique the authors use the Support Vector Machines (SVM). While these approaches demonstrate proof-of-concept feasibility, they are not yet suitable for real-time polyp detection due to high false-positive rates or low real-time performance Nie et al. (2024).

More recent work on polyp detection focuses on adapting state-of-the-art (SOTA) object detectors like YOLO or transformer-based models, making them suitable for colonoscopy data. These models integrate candidate generation, feature extraction, and classification into a single end-to-end system, overcoming many limitations of handcrafted features or the traditional two-stage CAD pipeline.

3 Object Detection Background

Within the following sections, the core principles on which modern object detection systems are built are introduced. Representative architectures and detection fundamentals are described and recent methodological developments relevant to real-time applications are summarized. Established evaluation strategies are then outlined to provide a common basis for assessing detection performance across different models and datasets.

3.1 Foundations of Object Detection

The fundamental idea of detection is identifying what is present in an image and determining where it appears. Given an input image I, the detector predicts detections $D = \{(b_i, c_i, p_i)\}$. Here, b_i represents the bounding box (bbox) coordinates, c_i the class label and p_i the confidence score Liu et al. (2020). Unlike classification, where a single label is assigned to an image, detection must process a variable number of instances and resolve spatial ambiguities. The main output of a detector are bounding boxes. These are axis-aligned rectangles that describe the position and size of an object in the image. They can either be represented by their corner coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$, or by the box center and its spatial dimensions (x, y, w, h), where (x, y) is the center position and w, h the width and height (Redmon et al., 2015; Liu et al., 2016).

From Hand-Crafted to Learned Features Before deep learning, object detectors used hand-crafted features processed by standard machine learning classifiers. Haar-like features, for example, calculate the difference in summed pixel intensities between adjacent rectangles and are first paired with AdaBoost for real-time face detection (Viola and Jones, 2001). Histogram of Oriented Gradients (HOG), on the other hand, first splits images into cells, and then uses the distribution of gradient orientations to capture local shapes. These are introduced in combination with linear Support Vector Machines (SVMs) for pedestrian detection (Dalal and Triggs, 2005). The Deformable Part Model (DPM) (Felzenszwalb et al., 2010) further improves flexibility through splitting an object in several parts and determining their spatial location relative to a root filter. The rise of convolutional neural networks (CNNs) Lecun et al. (1998) marks an important shift, as architectures originally developed for image classification now learn hierarchical feature representations directly from data. That idea can effectively be transferred to different visual domains Goodfellow et al. (2016). Building on this, the Region-based CNN (R-CNN) Girshick et al. (2014) adapts these deep architectures for object detection by applying a CNN to region proposals. Replacing hand-crafted descriptors with learned convolutional features led to substantially higher accuracy than Deformable Part-based Models on the PASCAL VOC benchmark (Everingham et al., 2010; Girshick et al., 2014).

3.2 Deep Learning based Object Detection

With the transition to learned features, the field of object detection has advanced to distinct architectural families. As outlined by Sun et al. (2024) modern deep detectors are broadly categorized into CNN-based models and Transformer-based models. The CNN-based approaches can further be divided into Region Proposal frameworks (Two-Stage) and Regression/Classification-based frameworks (One-Stage).

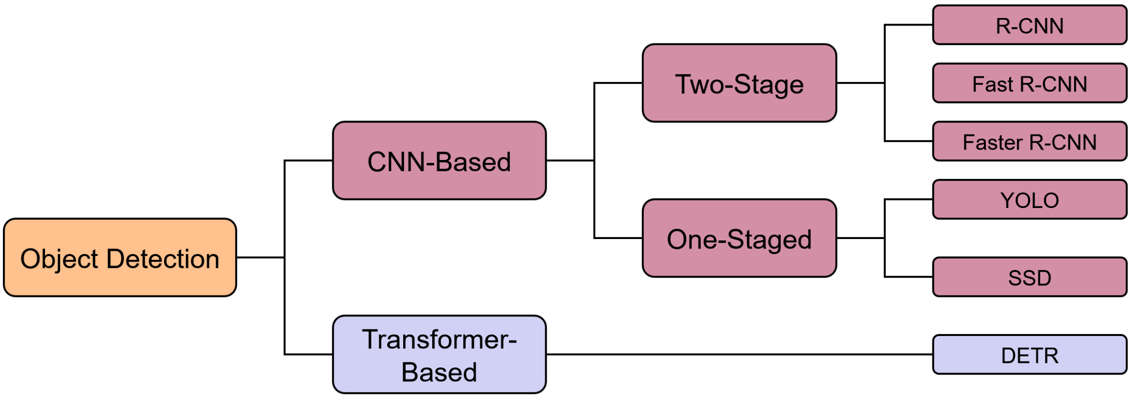


Figure 3: Categorization of selected Object Detection architectures used in this Thesis.

3.2.1 Two-Stage Detectors

R-CNN The R-CNN architecture, seen in Figure 4, operates by extracting about 2,000 class-independent region proposals of each image using selective search Uijlings et al. (2013). This search algorithm first segments the image into many small regions. It then iteratively merges adjacent regions based on similarity metrics such as color, texture, fill and size. The bottom-up approach captures potential objects at various scales effectively while at the same time reduces the search space compared to the previous exhaustive sliding-window method Uijlings et al. (2013). These proposals are then warped to a fixed resolution (227 × 227 pixels) to be compatible with the CNN input requirements, regardless of their original aspect ratio Girshick et al. (2014). The CNN then extracts a feature vector for each region and multiple pre-trained, class-specific linear SVMs classify the content of each of the regions. In this stage, (greedy) Non-Maximum Suppression (NMS) Neubeck and Van Gool (2006) is applied to filter out redundant, overlapping regions with lower confidence. Finally, a linear regression model refines the localization of the surviving proposals. By mapping learned features to numerical offsets, the model adjusts the coarse initial guess to match the ground-truth coordinates. Optimization is achieved by minimizing the difference between these predicted offsets and the targets. In the case of R-CNN, the model uses Ridge Regression, which adds a regularization term to the standard L_2 loss (Squared Error) to prevent overfitting by penalizing large errors heavily Girshick et al. (2014).

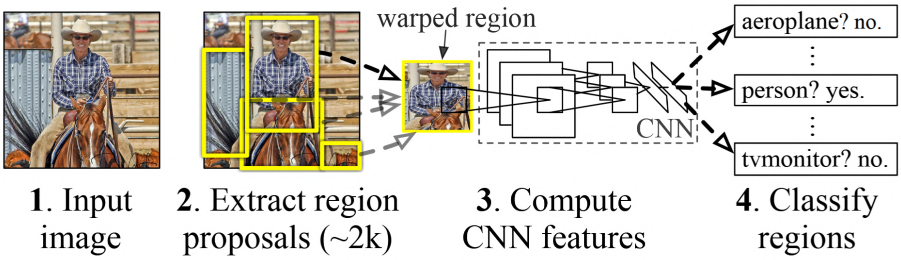


Figure 4: The R-CNN architecture by Girshick et al. (2014).

This approach shows promising results yet some limitations remain. One of the most concerning features is the fixed input size of 227×227 pixels. Warping proposals to such size often distorts the shape leading to quality loss and reduced accuracy Sun et al. (2024).

To mitigate this, He et al. (2014) proposed the SPP-Net, which uses the concept of Spatial Pyramid Matching (SPM) Lazebnik et al. (2006) into CNN architectures. SPM works by "separat[ing] the images into several scales from finer to coarser levels, and then aggregat[ing] local features into higher-level representations" Sun et al. (2024). Implementing this as a Spatial Pyramid Pooling (SPP) layer allows the network to *pool*, so to aggregate multiple feature maps into multi-level spatial bins (e.g., 1×1, 2×2, 4×4) and generate a fixed-length output vector for each image He et al. (2014). This eliminates the need for fixed-length input sizes and warping.

Fast R-CNN The improvements of SPP-Net still do not solve the multi-stage nature of R-CNN and the feature extraction was not yet fully end-to-end Zhao et al. (2019b). Fast R-CNN Girshick (2015) refines the SPP concept into a Region of Interest Pooling layer. The model, seen in Figure 5, takes region proposals from Selective Search as input and generates a feature map of the entire image using convolutional layers. The pooling layer extracts a fixed-size feature vector for each proposal directly from this shared map Zhao et al. (2019b). These vectors are then forwarded into fully connected layers that split into two output branches. First is a softmax layer that outputs region probabilities for $K + 1$ categories (the object classes and a background class). The other branch consists of a regression layer that outputs bounding box offset coordinates for each class Girshick (2015). This architecture enables the entire network to be trained end-to-end using a multi-task loss L Zhao et al. (2019b). Girshick defines the loss L as the sum of the classification loss (L_{cls}) and the localization loss (L_{loc}):

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \tag{1}$$

Here, the log loss $L_{cls}(p, u)$ is computed for the true class u based on the predicted discrete probability distribution p Girshick (2015). The term $[u \geq 1]$ is an indicator function that ignores the regression loss for the background ($u = 0$) Girshick (2015). The localization loss L_{loc} formulates a bounding-box regression as a smooth L_1 loss over the four box parameters. Given a regression target v and a predicted tuple t^u, the localization loss is computed over the four coordinates $i \in \{x, y, w, h\}$. Girshick (2015) defines this as:

$$L_{\mathrm{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \mathrm{smooth}_{L_1}(t_i^u - v_i) \tag{2}$$

where the smooth term is defined based on the error x:

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{3}$$

While R-CNN relies on squared error, this approach uses the L_1 loss (Absolute Error), which offers better robustness by computing the sum of absolute differences Girshick (2015). The quadratic term ensures differentiable and stable convergence for small errors, while the linear term limits the gradient magnitude for large errors. This prevents the exploding gradient problem caused by outliers often seen when using purely squared error (L_2) Girshick (2015). This optimization results in 213× increased detection speed in contrast to R-CNN while also resulting in better accuracy Sun et al. (2024).

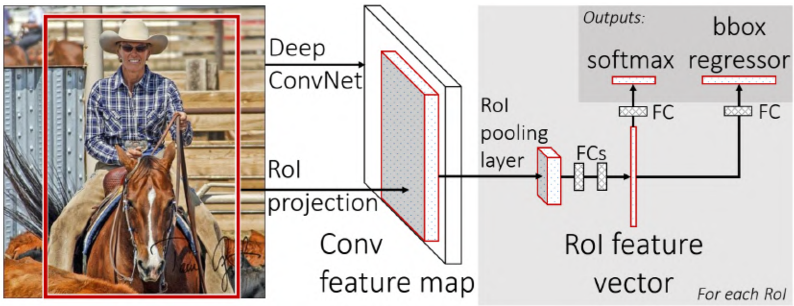


Figure 5: The Fast R-CNN architecture by Girshick (2015).

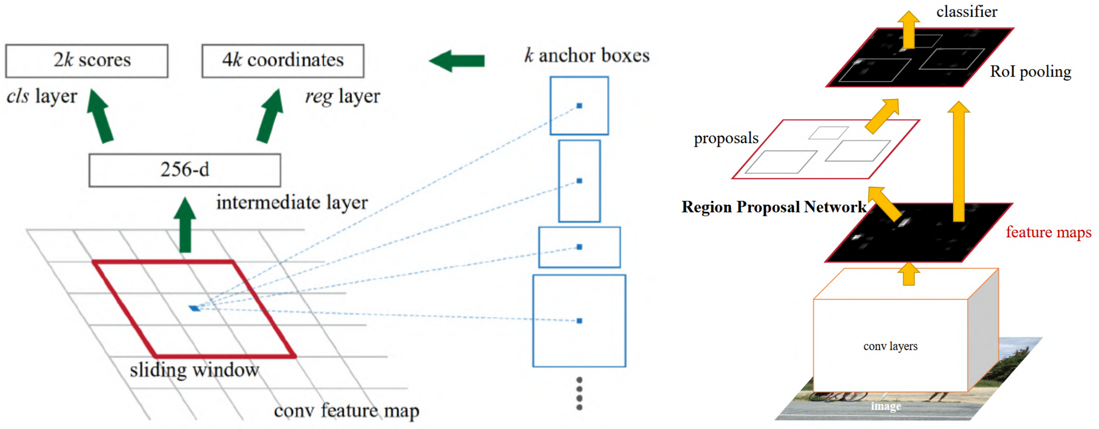
Faster R-CNN Previous models still rely on external algorithms like Selective Search for generating region proposals. Since these algorithms run as separate modules outside the convolutional network, the process is computationally intensive. They effectively become the new bottleneck for fast inference Ren et al. (2016); Zhao et al. (2019b). Ren et al. (2016) introduces Faster R-CNN to solve this problem by integrating proposal generation directly into the network architecture. It introduces the Region Proposal Network (RPN), a fully convolutional network that shares full-image convolutional features with the detection network, “which has the ability to predict object bounds and scores at each position simultaneously” Zhao et al. (2019b). As shown in Figure 6, the RPN uses a sliding window approach to scan over the feature map.

A key concept introduced by the RPN are anchors. Instead of using an image pyramid to handle scale, the RPN introduces reference boxes (anchors) at each sliding window position on the shared feature map. These anchors have fixed, pre-defined scales (e.g. 128×128, 256×256, 512×512 pixels) and aspect ratios (e.g. 1:1, 1:2 or 2:1), which allows objects of various shapes to be effectively taken into account Ren et al. (2016). For every anchor, the RPN predicts an objectness score (probability of an object vs. background) and a set of regression offsets that refine the anchor relative to its coordinates Ren et al. (2016). A single image produces thousands of anchors, most of which correspond to the background. Faster R-CNN does not compute the loss over all anchors. Doing so would cause the gradients to be dominated by negative samples and make optimization unstable Ren et al. (2016). Instead, the RPN draws a fixed-size mini-batch of anchors (typically 256) with a controlled foreground–background ratio, ensuring that informative positive anchors contribute meaningfully to the update Ren et al. (2016).

For each image, the RPN optimizes a multi-task loss function L that combines both classification and bounding box regression. Ren et al. (2016) define this as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{4}$$

The i refers to an index of a single anchor in the mini-batch. The term p_i denotes the predicted objectness probability, while p_i^* is the ground-truth label (1 for a positive anchor, 0 for background) Ren et al. (2016). The vectors t_i and t_i^* hold the four parameterized offsets used to refine each anchor towards the assigned ground-truth box. The classification loss L_{cls} is the two-class log loss, and the regression loss L_{reg} is the Smooth L_1 function adopted from Fast R-CNN Ren et al. (2016). The factor p_i^* ensures that box regression is applied only to positive anchors. Both terms are normalized by N_{cls} and N_{reg}, respectively, and weighted by the parameter λ to keep the objectives on comparable scales Ren et al. (2016).



(a) RPN

(b) Architecture

Figure 6: Region Proposal Network (a) within the Faster R-CNN architecture (b). Source: Ren et al. (2016).

The proposals produced by the RPN are filtered using Non-Maximum Suppression to remove strong overlaps. Only a small set of top ranked regions is forwarded to the

Fast R-CNN detection head. The entire system can be trained end-to-end using a multi-task loss similar to Fast R-CNN, combining classification and regression terms for both the RPN and the final detector. By sharing the convolutional backbone between the proposal generation and the detection, the cost of computing proposals becomes "nearly cost-free" (Ren et al., 2016).

Yet, this is not enough to effectively handle objects with multiscale features or small objects Sun et al. (2024). To overcome this, the Feature Pyramid Network (FPN) was introduced Lin et al. (2017). It augments the standard bottom-up feature hierarchy, where deep layers have high semantic information but low spatial resolution, with a top-down pathway Sun et al. (2024). This pathway propagates the rich semantic context from the deeper layers to the shallow, high-resolution layers through lateral connections. This generates a feature pyramid where every level is enhanced with strong semantics and fine localization details. The FPN can be trained end-to-end while not relying on a specific backbone and can be used for multiple other computer vision tasks Zhao et al. (2019b).

3.2.2 One-Stage Detectors

Two-stage methods, despite their gains in accuracy, are always bound to a multi-component pipeline. They rely on region proposal generation, feature extraction, classification, and bounding box regression Zhao et al. (2019b). Even the move toward end-to-end training often required detailed, alternative training steps to share convolutional parameters between the RPN and the main detection network. These limitations led to the development of one-step frameworks. These methods do not have a region proposal stage, but instead treat the entire detection process as a single global regression or classification task Zhao et al. (2019b) enabling real-time performance. Mapping is done directly from the input image to the final output of class probabilities and bounding box coordinates, reducing the computational cost.

You Only Look Once — YOLO(v1) This architecture was first introduced by Redmon et al. (2015) and quickly gained popularity as an efficient single-stage detector. The original version redefined object detection as a single, unified regression task performed directly on full images.

An overview of the model is given in Figure 7. It operates by using the full image directly. This image is first divided into an $S \times S$ grid, where each grid cell is made responsible for predicting the object whose center falls within its bounds (Redmon et al., 2015). Every part of the grid then predicts B bounding boxes, defined by the center coordinates (x, y) and dimensions (w, h), along with a corresponding confidence score. This score is a product of the object probability and the prediction quality, calculated as $Pr(\text{Object}) \times IOU_{\text{pred}}^{\text{truth}}$ (Redmon et al., 2015). Additionally, each cell predicts conditional class probabilities ($Pr(\text{Class}_i|\text{Object})$). During inference, the final, class-specific confidence for a box is determined by multiplying the conditional class probability by the confidence score of the box, resulting in: $Pr(\text{Class}_i) \times IOU_{\text{pred}}^{\text{truth}}$ (Redmon et al., 2015).

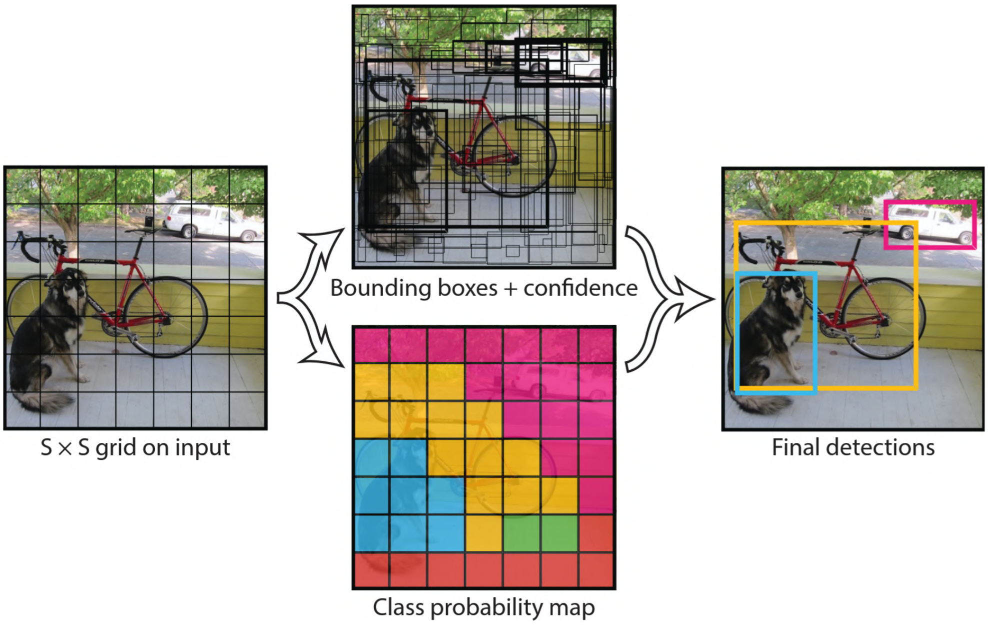


Figure 7: The YOLO object detection System by Redmon et al. (2015).

The original YOLO network architecture (Figure 8) is based on a modified version of GoogLeNet and consists of 24 convolutional layers followed by two fully connected (FC) layers Redmon et al. (2015). The network replaces Inception modules with 1 × 1 convolutions to reduce channel dimensions and 3 × 3 convolutions to extract spatial features. After the final convolutional block, the feature map is flattened and passed through the fully connected layers, which output the final $S \times S \times (B \cdot 5 + C)$ predictions tensor, where C is the class and each of the B bounding boxes contributes five numerical values (x,y,w,h,confidence) Redmon et al. (2015).

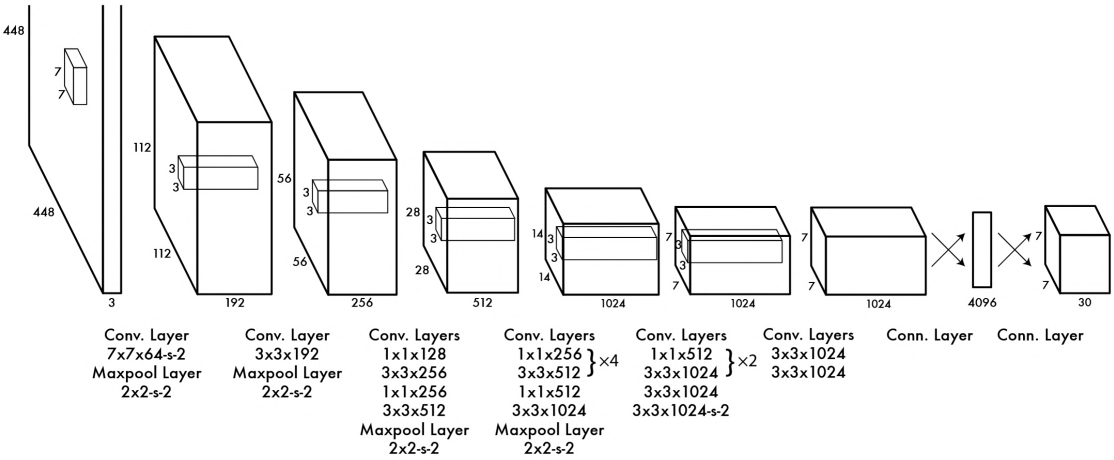


Figure 8: The YOLO Architecture (Redmon et al., 2015).

The model is optimized using a multi-part loss function based on the sum-squared error that minimizes coordinate, dimension, confidence, and classification errors. Redmon et al. (2015) define this as:

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&\quad + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
&\quad + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
&\quad + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
&\quad + \sum_{i=0}^{S^2} \mathbb{I}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{5}$$

This function selectively penalizes errors using the indicator function $\mathbb{I}_{ij}^{\text{obj}}$. Using the normalized bounding boxes (x,y,w,h) it ensures that only the bounding box predictor with the highest IOU (the 'responsible' one) is penalized for localization and object confidence errors (Redmon et al., 2015). C and $\hat{C}_i$ represent the predicted and ground-truth confidence. To counter the imbalance from numerous background regions, the hyperparameter λ_{noobj} is applied to decrease the loss from boxes without objects (set to 0.5). In contrast, λ_{coord} amplifies the localization error which tends towards zero if many grid cells do not contain anything (Redmon et al., 2015). The final term penalizes classification error $(p_i(c))$ only for the grid cells that contain an object ($\mathbb{I}_{i}^{\text{obj}}$ is true) (Redmon et al., 2015).

While this model is quite efficient, there are several limitations that remain. The main drawback is the difficulty in locating small or adjacent objects, as each grid cell can only represent one class, resulting in more localization errors than other detectors Redmon et al. (2015). Furthermore, because the model learns to predict bounding boxes directly from data, it produces relatively coarse features which is caused by the multiple downsampling layers in its architecture. This reliance on coarse features means YOLO "struggles to generalize to objects in new or unusual aspect ratios or configurations" Redmon et al. (2015).

Single Shot MultiBox Detector — SSD This Detector, introduced by Liu et al. (2016), is a single-stage model designed to handle the limitations of YOLOv1 while at the same time maintaining its speed. Additionally, its accuracy remains comparable to, or even surpasses, the slower two-stage methods like Faster R-CNN.

The SSD architecture begins with a modified version of VGG-16 as the backbone, truncated before its final classification layers (Liu et al., 2016). This backbone produces the first set of feature maps. On top of it, several additional convolutional feature layers are attached which decrease in spatial resolution, giving SSD access to feature maps that become progressively coarser (Liu et al., 2016). Unlike the limited, fixed grid of YOLOv1, SSD introduces a predefined set of default boxes (similar to anchors in the RPN) that have different scales and ratios and tile every location across these multiple feature maps (Liu et al., 2016). For each box, the model predicts the shape offsets, determining how the box should shift and resize to better match an object, and the class scores for all object categories (Liu et al., 2016). These predictions come from small 3×3 convolutional filters applied across the feature maps.

During training, SSD matches default boxes to ground-truth objects based on IoU overlap Liu et al. (2016). Boxes that overlap sufficiently become positives, and all remaining boxes act as negatives. A combined loss then optimizes both bounding-box regression and classification. To ensure stable training despite a large number of negative (background) predictions, SSD employs two core strategies: Hard-negative mining that keeps the ratio between positive and negative examples balanced, and extensive data augmentation that helps the model handle objects at different scales (Liu et al., 2016). At inference time, the network generates all predictions in one shot, and a class-wise non-maximum suppression step produces the final set of detections, filtering out redundant predictions.

3.2.3 Transformer-based Detectors

The dominance of CNN-based models in object detection was reexamined after the success of the Transformer architecture in natural language processing (NLP) Vaswani et al. (2017); Sun et al. (2024). Despite efficiency gains from single-stage detectors, the entire field still relies on complex pre-knowledge elements such as predefined anchor boxes and a labor-intensive post-processing step, non-maximum suppression (NMS), to filter redundant bounding boxes.

DEtection TRansformer — DETR Carion et al. introduces an end-to-end object detector that removes components like anchors or postprocessing steps like NMS. As stated by the authors, two elements are central to this formulation: a loss for one-to-one matching between predictions and ground-truth boxes, and an architecture capable of producing a set of predictions that can model their relations. For the matching, DETR computes an optimal bipartite assignment between the N predictions $\hat{\mathbf{y}}$ and the ground-truth set $\mathbf{y}$, where $\mathbf{y}$ is padded to size N with the no-object symbol $\varnothing$ (Carion et al., 2020). The optimal assignment $\hat{\sigma}$ is obtained by minimizing a pair-wise matching cost over all permutations $\mathfrak{S}_N$:

$$\hat{\sigma} = \arg \min_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\mathbf{y}_i, \hat{\mathbf{y}}_{\sigma(i)}) \tag{6}$$

This optimal assignment is computed efficiently with the Hungarian algorithm (Carion et al., 2020). The matching cost accounts for both class prediction and bounding-box similarity (Carion et al., 2020). Given $\hat{\sigma}$, DETR optimizes the Hungarian loss

$$\mathcal{L}_{\text{Hungarian}}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{N} \Big[-\log \hat{P}_{\hat{\sigma}(i)}(c_i) + \mathbb{I}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(\mathbf{b}_i, \hat{\mathbf{b}}_{\hat{\sigma}(i)}) \Big], \tag{7}$$

where the box loss $\mathcal{L}_{\text{box}}$ combines L_1 regression and generalized IoU, and box regression is only applied for matched object slots ($c_i \neq \varnothing$) (Carion et al., 2020). Unmatched slots are trained toward the no-object class through the classification term (with a reduced weight for the no-object class imbalance in practice) (Carion et al., 2020).

The architecture, seen in Figure 9, consists of three components. First, a CNN backbone extracts a spatial feature map. Before the transformer encoder processes the features, DETR reduces the channel dimension with a 1×1 convolution, flattens the spatial map into a sequence, and adds fixed position encodings to this input (Carion et al., 2020). The encoder then applies multi-head self-attention to aggregate information across all spatial positions. Both encoder and decoder consist of six identical layers that are cascaded Yao et al. (2021).

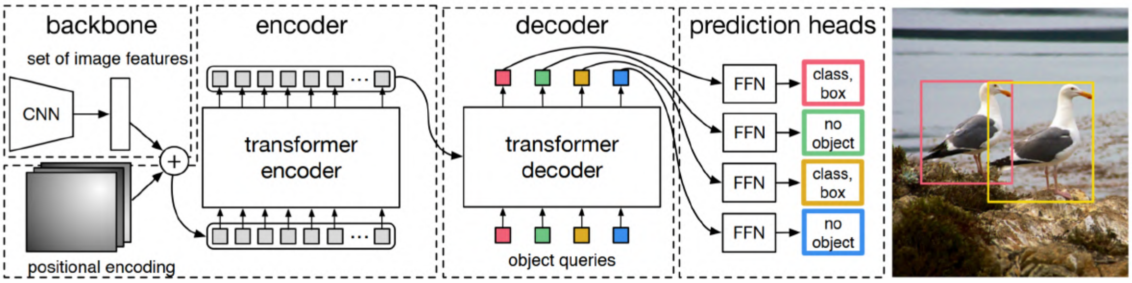


Figure 9: DETR Architecture as given in Carion et al. (2020).

The transformer decoder receives a fixed set of learned object queries, each representing one potential detection (Carion et al., 2020). These queries are distinct by design and are combined with positional information before entering every decoder layer. Through repeated self-attention and encoder–decoder attention, the decoder transforms the queries into N output embeddings, making the model reason about all objects using pairwise relations and the full image context (Carion et al., 2020). Each decoder output is passed through a small three-layer feed-forward network that predicts a class label and normalized box coordinates for all objects independently at the same time (Carion et al., 2020).

While improving accuracy compared to Faster-RCNN, the initial DETR model has its own set of limitations. The authors report a slow convergence, requiring about 300–500 epochs for the model to train efficiently on the COCO dataset and point out its limitation for detecting small objects. Several adaptions have been implemented improving on the previous drawbacks. Deformable DETR for example, transforms its multi-scale attention modules to “only attend to a small set of key sampling points around a reference” (Zhu et al., 2020). The result is a more efficient training requiring ten times less epochs for convergence while also improving accuracy on smaller objects (Zhu et al., 2020). Another variant is Efficient DETR, which reduces the number of encoder layers to three and the decoder layers to just one, while maintaining the same accuracy and reducing the epochs for convergence to just 36 (Yao et al., 2021). Real-Time DETR (RT-DETR) further improves the efficiency Zhao et al. (2024). A more detailed explanation is given in Section 3.3.2.

3.2.4 Improved Concepts

IoU Variants Much of the recent progress in object detection is driven by refinements inside existing architectures rather than entirely new designs. To evaluate the quality of a prediction, the Intersection-over-Union (IoU) is used. It measures the geometric overlap between a predicted box B_p and a ground-truth box B_g:

$$\mathrm{IoU}(B_p, B_g) = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \tag{8}$$

A prediction is considered a True Positive (TP) if the predicted class matches the ground truth and the IoU exceeds a predefined threshold, typically 0.5 (Liu et al., 2020). Although IoU provides an intuitive measure of spatial agreement, it is not well suited as a direct regression loss because the gradients vanish when predicted and ground-truth boxes do not overlap (Rezatofighi et al., 2019). Various IoU variants were introduced to solve this problem. Generalized-IoU (GIoU) improves IoU by adding the smallest enclosing box around the two regions, providing meaningful gradients even when they do not overlap (Rezatofighi et al., 2019). Distance-IoU (DIoU) optimizes “the normalized distance between central points of two bounding boxes” (Zheng et al., 2019), so the centers of the predicted and ground-truth boxes, encouraging faster convergence and smaller regression errors, particularly when boxes are far apart. Finally, Complete-IoU (CIoU) improves upon DIoU by adding an aspect-ratio consistency term to penalize differences in box shapes, leading to more reliable regression when overlap, center distance, and geometry must be optimized jointly (Zheng et al., 2019).

Feature Refinement In object detection architectures, feature extraction is typically done within the backbone network and an optional neck module. The backbone is a convolutional neural network responsible for extracting hierarchical visual features from the input image, while the neck aggregates and refines these features across multiple scales before they are passed to the detection head. A commonly used neck is the Feature Pyramid Network (FPN) introduced a top-down pathway with lateral skip-connections, enriching high-resolution layers with strong semantic context (Lin et al., 2017). Path Aggregation Network (PANet) extends this idea with an additional bottom-up path, improving information flow toward lower levels used for small object detection (Liu et al., 2018). More recent search-based variants such as Neural Architecture Search - FPN (NAS-FPN) use neural searches to repeatedly stack and optimize cross-scale fusion blocks, often improving accuracy at significant computational cost (Ghiasi et al., 2019). By contrast, BiFPN adopts a weighted bidirectional structure that shares features across scales more efficiently, allowing models to scale from lightweight real-time variants to high-accuracy configurations Tan et al. (2020).

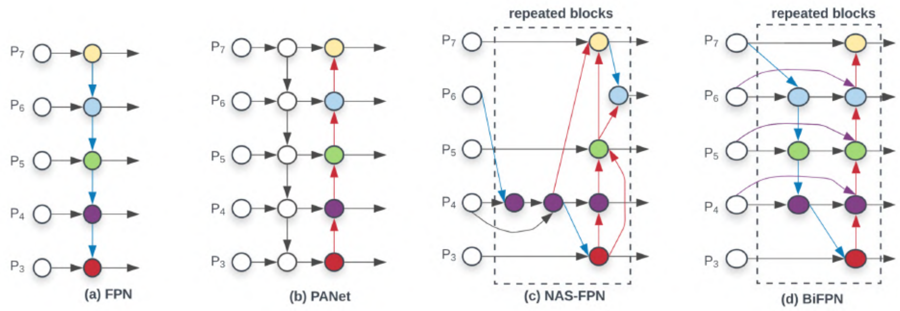


Figure 10: Summary of Feature Extractors by Tan et al. (2020).

3.3 Improved Detection Architectures

While the previous section detailed the foundational methods, current state-of-the-art methods represent optimized and improved versions that achieve better results on standard benchmarks like COCO compared to their base architectures. This section explains some of these advanced detection architectures, specifically the latest iterations of the You Only Look Once (YOLO) family and the efficient Real-Time DETR (RT-DETR), which are then evaluated in this thesis for their applicability to medical image analysis.

3.3.1 YOLOv11

Over the past decade, the YOLO family has undergone continuous development, with successive revisions improving feature extraction, multi-scale prediction, loss formulations, and training strategies. An overview of these architectural and methodological developments is provided in Appendix A.1. This thesis focuses on recent Ultralytics implementations of YOLOv8 and YOLOv11, with focus on YOLOv11 as the most recent iteration evaluated in this work (Jocher and Qiu, 2024).

Similar to earlier variants, YOLOv11 follows a single-stage detection architecture consisting of a backbone for hierarchical feature extraction, a neck for multi-scale feature aggregation, and a detection head for prediction of object locations and classes. Given an input image, the backbone first applies a sequence of convolutional layers, each performing a two-dimensional convolution followed by batch normalization and a SiLU activation function (Hidayatullah et al., 2025). These reduce the spatial resolution while increasing the number of feature channels. Early stages extract low-level visual patterns such as edges and textures and transform the input into a compact feature representation. Feature refinement at fixed spatial resolutions is performed using the C3k2 block, which extends the C2f module introduced in YOLOv8. The C3k2 block first applies a 1×1 convolution to project the input features into an intermediate representation, followed by a channel-wise split into two equal parts. One part forms a bypass branch that keeps the original information flow, while the other part is processed sequentially by a stack of feature extraction blocks. In the case of smaller model variants (n and s), these blocks correspond to standard bottleneck layers composed of two 3×3 convolutions with optional residual connections, resulting in behavior of the original C2f. For larger variants (m, l, and x), the bottleneck layers are replaced by C3k blocks, which follow a Cross Stage Partial design for more precise spatial feature modeling (Hidayatullah et al., 2025). The bypassed features and all intermediate outputs from the processed branch are concatenated along the channel dimension and fused by a final 1×1 convolution. Deep in the backbone, YOLOv11 adds the Spatial Pyramid Pooling - Fast (SPPF) module to aggregate contextual information at multiple scales. SPPF applies repeated max-pooling operations to the same feature map and concatenates the resulting representations, enabling the network to combine both local and global context without altering the

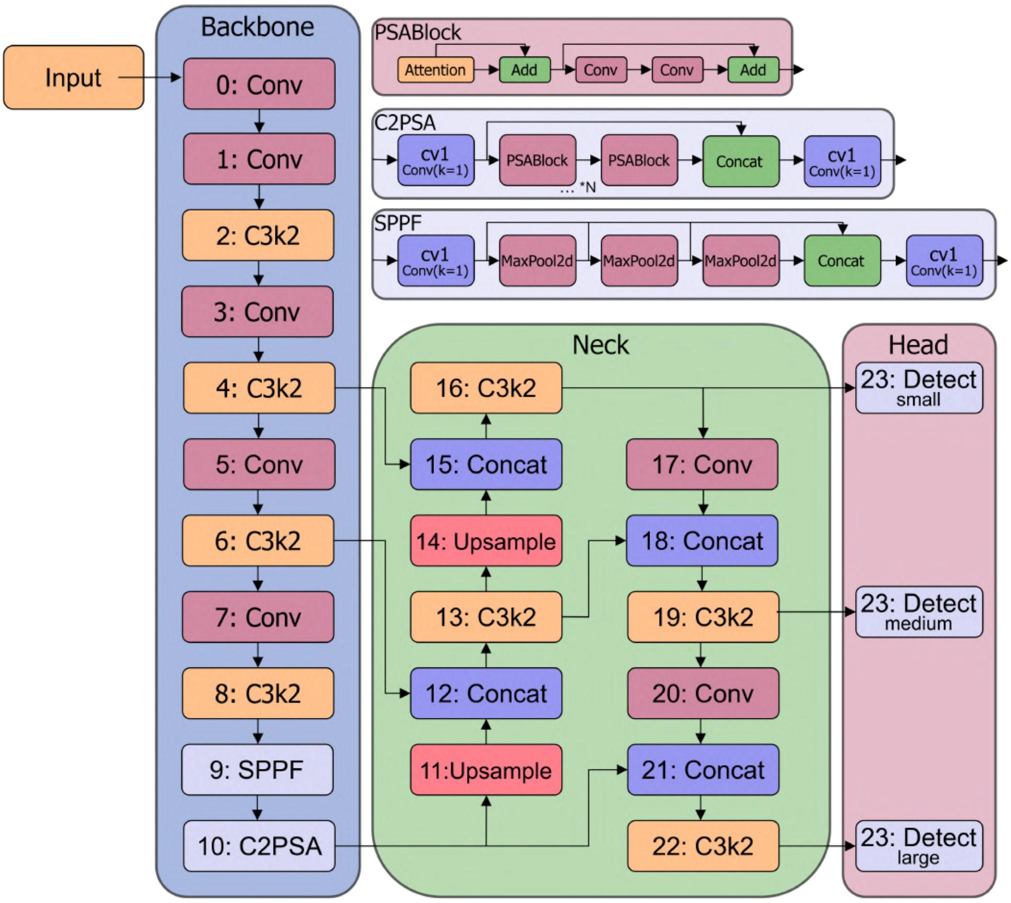


Figure 11: Overview of the YOLOv11 architecture.

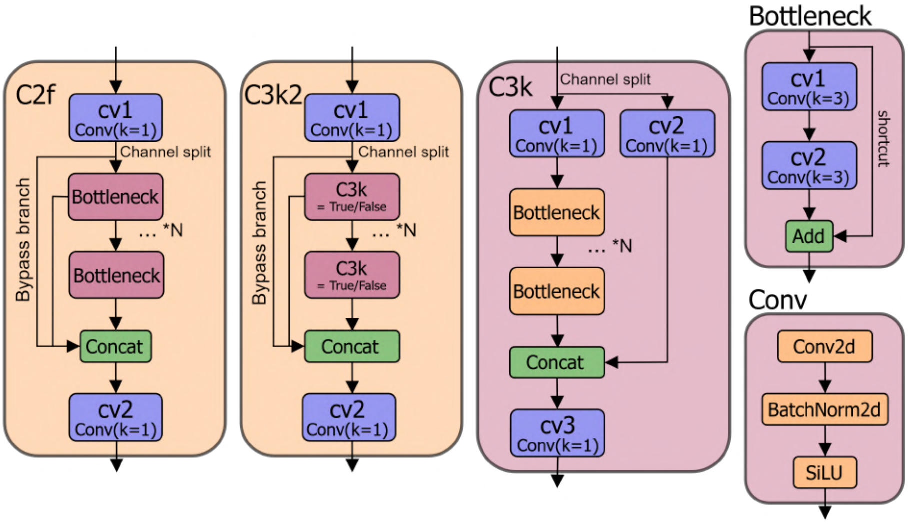


Figure 12: Conceptual comparison of the C2f block used in YOLOv8 and the C3k2 block employed in YOLOv11.

spatial resolution. A C2PSA block is followed and used for global reasoning through self-attention with position information. Like other CSP modules, C2PSA splits the input features into a bypass- and a processed branch. The processed branch is refined using one or more Attention blocks, each combining multi-head self-attention with a small feed-forward network and residual connections. Using the attention mechanism improves detection of occluded or small objects (Sapkota et al., 2025).

The neck of YOLOv11 aggregates features across multiple spatial scales to support detection of objects with varying sizes. As in YOLOv8, the neck is based on Feature Pyramid Networks (FPN) and Path Aggregation Networks (PANet). These provide top-down and bottom-up pathways for combining high-level semantic information with spatial details (Terven et al., 2023). High-level, low-resolution feature maps are upsampled using nearest-neighbor interpolation and concatenated with lower-level backbone features of matching spatial dimensions. Then, additional convolutional layers refine the fused representations.

The decoupled detection head translates the neck features into final predictions. Recent YOLO versions use fully anchor-free heads that directly predict object centers, the resulting box dimensions, and class probabilities (Sapkota et al., 2025; Terven et al., 2023). Outputs are generated at several resolutions, allowing the model to predict for small, medium, and large objects within a single forward pass.

However, still YOLOv11 has several open challenges. The review by Sapkota et al. (2025) reports that YOLOv11 still struggles with very small or strongly rotated objects. The authors also note a tendency toward overfitting when trained on limited or low-diversity datasets, which may affect generalization. Finally, YOLOv11 maintains the characteristic speed–accuracy trade-off of one-stage detectors: while highly efficient, its accuracy may degrade in visually complex scenes (Sapkota et al., 2025).

3.3.2 RT-DETR

Although DETR offers a good foundation, Zhao et al. point out several issues that limit its applicability for real-time applications, including slow training convergence, high computational cost and hard-to-optimize queries. In particular, when DETR processes long multi-scale feature sequences, the encoder becomes a computational bottleneck (Zhao et al., 2024).

To address these limitations, Zhao et al. introduce a real-time end-to-end variant of DETR. Its overall structure, shown in Figure 13, remains consistent with the DETR family and includes a backbone, an encoder, a decoder and a set of object queries. The key differences lie in the redesigned encoder and the way queries are initialized.

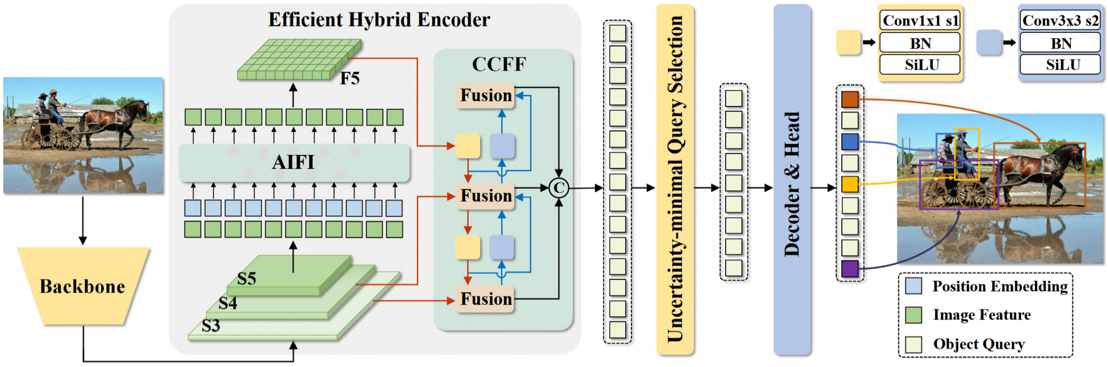


Figure 13: The adapted RT-DETR Architecture by Zhao et al. (2024).

The original transformer encoder is replaced by an efficient hybrid encoder designed to process multi-scale features without the heavy computational cost of global attention over long sequences (Zhao et al., 2024). RT-DETR achieves this with two components. The Attention-Based Intra-Scale Feature Interaction (AIFI) module performs self-attention only on the highest-level feature map (S_5), rather than on concatenated multi-scale features (Zhao et al., 2024). The authors argue that applying self-attention on high-level features with richer semantic concepts is sufficient for object-level interactions, while avoiding attention on lower-level features reduces latency and can even slightly improve accuracy. The Cross-Scale Feature Fusion (CCFF) module then merges information across scales by fusing $\{S_3, S_4, F_5\}$ using convolutional fusion blocks with 1×1 channel alignment, RepConv-based RepBlocks for feature interaction, and element-wise addition (Zhao et al., 2024). A second modification concerns query initialization. Building on prior DETR variants that initialize queries from encoder features ranked by classification confidence, RT-DETR proposes an “uncertainty-minimal query selection” scheme (Zhao et al., 2024). It considers both classification and localization confidence to rank encoder features and selects the most certain candidates to initialize the decoder queries. This provides higher quality initial queries for the decoder and leads to improved accuracy (Zhao et al., 2024).

The decoder follows DETR by predicting class labels and bounding boxes for a fixed set of N object queries (typically N=300) for every input image. Due to the one-to-one bipartite matching used during training, the model learns to assign high confidence scores only to the specific queries that successfully localize a target object. The remaining "junk" queries are actively suppressed to the background class with confidence scores approaching zero. During testing, this allows for a clean prediction output. By applying a specific confidence threshold, the many near-zero background predictions are eliminated, leaving only the usable detections. This internal negotiation between queries in the decoder's self-attention layers ensures that spatial redundancy is handled without the need for Non-Maximum Suppression (NMS), maintaining an end-to-end pipeline that is both computationally efficient and highly accurate.

3.4 Benchmarking Datasets

To assess the quality and effectiveness of object detection models, their performance must be evaluated under standardized and reproducible conditions. In practice, this is done by benchmarking detectors on large, publicly available datasets, such as PASCAL VOC, ImageNet or Microsoft COCO. A key advantage of these datasets is the existence of predefined splits along with clearly specified metrics. This shared experimental setup allows results from different models to be compared directly, without the need to reimplement evaluation pipelines or debug incompatible protocols.

PASCAL VOC The PASCAL Visual Object Classes (VOC) benchmark was one of the earliest widely adopted datasets for object detection and classification. It contains images of common object categories such as animals, cars, and household items, annotated with bounding boxes and class labels (Everingham et al., 2010). Detection performance on PASCAL VOC is evaluated using average precision at a fixed intersection-over-union threshold of 0.5, averaged across classes (Everingham et al., 2010). It is not used as much anymore due to its limited size and amount of classes.

ImageNet This dataset was originally introduced for large-scale image classification, but was later extended to include object detection and localization tasks (Deng et al., 2009). ImageNet detection played an important role in early large-scale detection research and pretraining on ImageNet classification still remains a standard practice, as it provides robust visual representations that transfer well to downstream detection tasks.

Microsoft COCO The Microsoft Common Objects in Context (COCO) dataset represents the most influential benchmark for modern object detection. It contains over 200,000 images with more than one million annotated object instances across

80 categories, many of which appear in complex scenes with clutter, and strong scale variation (Lin et al., 2014). An official evaluation is given in the `pycocotools` library, which provides a reference for computing metrics such as (mean) average precision. This library takes the top 100 predictions per image (sorted by box confidence) and performs matching between predicted and ground-truth bounding boxes across multiple IoU thresholds (Lin et al., 2014). This ensures consistent and reproducible evaluation across different models and frameworks. COCO is also often used for pretraining object detection models.

3.5 Evaluation Metrics

Depending on how detector outputs are interpreted, object detection performance can be evaluated under different formulations. Detection-level metrics assess localization accuracy of individual bounding boxes. Frame-level alert metrics reformulate detection as a binary classification problem by collapsing detector outputs into a single decision per frame and event-level metrics aggregate detections across frames. Standard benchmarks such as COCO primarily rely on detection-level metrics (e.g., AP/mAP). In contrast, medical video analysis additionally requires metrics that characterize false-alarm behavior on polyp-negative frames, temporal aggregation across frames, and real-time feasibility.

3.5.1 Detection-Level Metrics

Individual predicted bounding boxes are matched to ground-truth (GT) boxes using an Intersection over Union (IoU) criterion. A prediction is counted as a true positive (TP) if it matches a GT object with an IoU exceeding a specified threshold under a one-to-one matching policy, where each GT object can be matched to at most one prediction. Predictions that do not match any GT object are counted as false positives (FP), while GT objects without a matched prediction are counted as false negatives (FN). True negatives are not defined at the detection level, as the background does not create a finite, countable set of negative instances (Padilla et al., 2020).

(Mean Average) Precision and Recall Precision measures the proportion of predicted detections that are correct, while recall measures the proportion of GT objects that are successfully detected (Padilla et al., 2020). Both depend on the chosen confidence threshold. As modern detectors typically produce many low-confidence predictions, reporting performance at a single threshold is often not informative (Everingham et al., 2010). Average Precision (AP) addresses this by computing the area under the precision–recall curve obtained by sweeping the confidence threshold, while fixing the matching criterion via an IoU threshold (Padilla et al., 2020). Mean Average Precision (mAP) is then defined as the mean of AP values over all n object classes k, and depending on the benchmark, over multiple IoU thresholds. COCO reports $\mathrm{mAP}_{50:95}$, which averages AP across multiple IoU thresholds from 0.50 to 0.95 in steps of 0.05 (Lin et al., 2014).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad mAP = \frac{1}{n}\sum_{k=1}^{n} AP_k \tag{9}$$

Sensitivity Sensitivity is equivalent to recall and indicates how reliably objects are detected. Specificity is not defined at detection level for the same reason TN is undefined.

$$\text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

F-Scores The F_1 score ($\beta = 1$) weights precision and recall equally, whereas the F_2 score ($\beta = 2$) prioritizes recall. The latter is helpful in polyp detection, as missed lesions are typically considered more critical than additional false positives.

$$\mathrm{F}_\beta = (1 + \beta^2)\frac{\text{Precision} \cdot \text{Recall}}{\beta^2\,\text{Precision} + \text{Recall}}. \tag{11}$$

Free-response Receiver Operating Characteristic (FROC) The receiver operating characteristic (ROC) assigns a single decision to each image and therefore does not distinguish between one object or false alarm and multiple objects and false alarms within the same image (He and Frey, 2009; Chakraborty, 2013). FROC extends ROC by allowing an arbitrary number of detections per image. Each prediction is classified as a true positive if it falls within a predefined acceptance region around a GT box (such as IoU), while predictions that do not match any ground-truth object are counted as false positives (Chakraborty and Winter, 1990). Performance is characterized by the trade-off between detection sensitivity and the average number of false detections per image (He and Frey, 2009):

$$\text{FPs per image (FPPI)} = \frac{\text{number of false localizations}}{\text{number of images}}. \tag{12}$$

Since the mean number of false positives per image is an unbounded count, the FROC x-axis FPPI is not restricted (Chakraborty and Winter, 1990).

Alternative Free-response Receiver Operating Characteristic (AFROC) AFROC replaces the unbounded FPPI axis with the false positive fraction (FPF), defined as the fraction of negative images that have at least one false-positive decision above the operating threshold (Chakraborty and Winter, 1990; He and Frey, 2009):

$$\text{FPF} = \frac{\text{negative images with} \geq 1 \text{ FP decision}}{\text{negative images}} \in [0, 1] \tag{13}$$

This builds a curve within the unit square and supports ROC-like scalar summaries, such as the area under the AFROC curve, while retaining sensitivity on the ordinate (Chakraborty and Winter, 1990; Chakraborty, 2013).

3.5.2 Frame-Level Metrics

To characterize alert behavior in videos, detector outputs can be reduced to a binary decision per frame. A frame is predicted positive if it contains at least one detection exceeding a confidence threshold (while matching an IoU threshold when GT is present), and negative otherwise (Sokolova and Lapalme, 2009). This creates a standard confusion matrix at the frame level where TN becomes defined:

- $\mathrm{TP}_{\mathrm{frame}}$: GT-positive frame with at least one valid detection,
- $\mathrm{FN}_{\mathrm{frame}}$: GT-positive frame without a valid detection,
- $\mathrm{FP}_{\mathrm{frame}}$: GT-negative frame with at least one detection,
- $\mathrm{TN}_{\mathrm{frame}}$: GT-negative frame without detections.

Frame-level true positive rate (TPR) (sensitivity), false positive rate (FPR) and specificity are then:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \qquad \text{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}. \tag{14}$$

3.5.3 Runtime Metrics

Runtime is critical for real-time applications. The latency for processing a single image is commonly reported as the sum of preprocessing, inference, and postprocessing time. The Throughput is expressed in frames per second (FPS). Real-time operation typically requires at least 30 FPS or less than 33ms for the inference time (Pacal et al., 2022).

$$t_{\mathrm{img}} = t_{\mathrm{pre}} + t_{\mathrm{inf}} + t_{\mathrm{post}}, \qquad \mathrm{FPS} = \frac{1}{t_{\mathrm{img}}} \quad \text{or} \quad \mathrm{FPS} = \frac{1000}{t_{\mathrm{img}}\ [\mathrm{ms}]}. \tag{15}$$

4 Related Work

This chapter reviews prior research on automated polyp detection in colonoscopy, also including on how modern object detection methods have been applied in this field. It summarizes commonly used colonoscopy datasets and examines how existing studies adapt detection architectures and evaluation strategies to medical video data. Recurring design choices and evaluation practices are highlighted that shape reported performance, providing context for the methodological decisions made in this thesis.

4.1 Colonoscopy Datasets

Several datasets focus exclusively on still images with segmentation masks or bounding box coordinates. Examples for this are **ETIS-Larib**, which consists of 196 selected images annotated for segmentation (Silva et al., 2014). Similarly, **Kvasir-SEG** provides 1,000 polyp images at varying resolutions with masks (Jha et al., 2019). **CVC-ColonDB** has 300 polyp images extracted from multiple colonoscopy cases, all of which show polyp-positive frames only (Bernal et al., 2012). **CVC-ClinicDB** extends this list by providing 612 polyp images extracted from short video sequences, each annotated with masks (Bernal et al., 2015). **PICCOLO** includes a curated set of manually selected polyp images with segmentation masks with clinical metadata (Sánchez-Peralta et al., 2020). **CVC-ClinicVideoDB** contains positive and negative colonoscopy video clips with a total of 11,945 frames, with segmentation masks provided for every frame (Tajbakhsh et al., 2016). The **KUMC** dataset further consists of 37,899 manually selected colonoscopy video clips in which frames with motion blur, defocus or extreme illumination changes are removed (Li et al., 2021). Next, **LDPolypVideo** (Ma et al., 2021) and the **SUN colonoscopy** database (Misawa et al., 2021) provide annotated video clips centered around individual lesions, enabling limited temporal analysis while still restricting evaluation to short, curated sequences rather than full-length examinations.

A common characteristic across both frame-based and clip-based datasets is a strong imbalance toward positive samples. Many datasets either exclude non-polyp frames entirely or include them only in limited quantities. This design choice simplifies the problem, but it does not reflect real-world colonoscopy scenarios, where the majority of frames do not contain polyps Biffi et al. (2024). As a consequence, evaluation on such datasets tends to overestimate practical performance and provides limited insight into false positive rates, temporal consistency, and procedure-level behavior. To address these limitations, the **REAL-Colon** dataset was introduced as a large-scale, multi-center collection of full-length colonoscopy procedures with many frame-level bounding-box annotations Biffi et al. (2024). Therefore, REAL-Colon is adopted as the primary dataset in this thesis, while PICCOLO and SUN are considered as representative frame-based and clip-based benchmarks. A detailed description of these is given in Section 5

4.1.1 Evaluation Protocols

Reported results depend strongly on how datasets define splits, which frames are considered for evaluation, and how positives and negatives are constructed. In KUMC, multiple object detectors, including Faster R-CNN, SSD, YOLOv3, and YOLOv4, are evaluated across three complementary experiments (Li et al., 2021). First, a frame-based two-class detection task assesses localization and classification of adenomatous and hyperplastic polyps in individual frames. Second, a frame-based one-class detection task merges both polyp types into a single foreground category. Third, a sequence-based two-class classification task aggregates frame-level predictions across short video sequences using majority voting to produce a single prediction per lesion. Performance is reported using precision, recall, and F1-score. For the two-class frame-based task, YOLOv3 achieves the highest precision (74.9%), but with low recall (19.7%). In the one-class setting, YOLOv3 again achieves the highest precision (95.9%) while maintaining a recall of 78%, whereas YOLOv4 yields slightly lower precision but higher recall (47.8%) in the two-class task. All results are obtained at a fixed confidence threshold of 0.5, with IoU thresholds varying between 0.45 and 0.5 depending on the detector (Li et al., 2021).

SUN authors developed a YOLOv3-based detector. For training, non-publicly available images of colonoscopies are taken, consisting mainly of positive images and a limited number of negative images that are found to improve detection performance (Misawa et al., 2021). To mitigate overfitting, horizontal and vertical flipping, brightness adjustment, sharpness modification and L2 regularization during training is used. Sensitivity and specificity are computed based on binary classification of individual frames as polyp-present or polyp-absent. Lesion-level sensitivity is defined as the proportion of polyps for which more than half of the associated frames are correctly detected. A prediction is considered correct if the IoU exceeds 0.3, yet a value for the confidence is not given (Misawa et al., 2021). With their model they achieved a lesion sensitivity of 98% and per-frame sensitivity and specificity of 90.5% and 93.7%, respectively.

4.1.2 Benchmarks on Full-Length Colonoscopy Procedures

Beyond frame-based and short clip benchmarks, only few studies explicitly evaluate CADe systems or object detection models on full-length screening colonoscopies using clinically motivated operating points and procedure-level analyses.

The REAL-Colon paper uses a SSD detector with ImageNet-pretrained weights and applies random cropping, scaling and horizontal flipping to the image. In the first experiment, training data are constructed by varying the number of polyp frames and the proportion of negative frames. Based on validation performance on the test set, the best-performing configuration (using all positive and all negative images) is selected for all subsequent analyses. Detection performance is reported using $\mathrm{mAP}_{50:95}$ (0.216), as well as AP_{50} (0.338) and AP_{75} (0.245). In addition, event-level False Positive Rate (FPR) (0.054) and True Positive Rate (TPR) (0.505) are reported, defined as “the percentage of polyp-negative frames in which models erroneously detected bounding boxes and the percentage of positive frames in which models flagged an alert, respectively” (Biffi et al., 2024). Yet, a minimal confidence threshold or the IoU used for the predictions is not given. Further experiments evaluate detection performance across polyp subgroups defined by size, histology, and anatomical location. An additional analysis focuses on early detection by restricting evaluation to frames within the first one and three seconds of polyp appearance.

Another example is provided by Fitting et al. and Troya et al., who introduce the CADe system EndoMind and evaluate it on full-length colonoscopy videos using a dedicated benchmark dataset. EndoMind is based on a YOLOv5 architecture with additional post-processing, pretrained on still-image polyp datasets and further trained on frames extracted from routine colonoscopies. The evaluation spans 101 screening colonoscopies with frame-level annotations indicating polyp presence. Only screening procedures with at least one adenomatous polyp and adequate bowel preparation are included, while frames recorded during polypectomy, narrow-band imaging, documentation freezes, and rectal inspection are excluded (Troya et al., 2024). Performance is assessed at the frame level: a prediction is considered correct if a detected bounding box overlaps any ground-truth annotation (IoU>0), reflecting the objective of CADe systems to alert the endoscopist to the presence and approximate location of a lesion (Fitting et al., 2022). Predictions are filtered using a confidence threshold of 0.2. EndoMind is compared against several commercial CADe systems, including GI Genius (versions 1 and 2) and two variants of Endo-AID. While all systems detect nearly all polyps in at least one frame (98.92–100% sensitivity), substantial differences are observed in frame-level sensitivity and detection latency. EndoMind achieves a mean per-frame sensitivity of 60.22%, and a median first-detection time of 1083 ms. Inference speed, computational efficiency, and the impact of post-processing are not reported, and the benchmark relies on curated frame exclusions rather than unfiltered full-procedure recordings.

Lee et al. develop a real-time CADe system based on a YOLOv2 detector and validate it across multiple independent datasets comprising still images, edited video clips, and unaltered full-length colonoscopy videos. The detector is first evaluated on internal and public still-image datasets, followed by clip-based video evaluation using per-frame and per-polyp sensitivity as well as false-positive rates. To stabilize frame-level predictions, the authors apply temporal post-processing based on median filtering, aggregating detections over sliding windows of consecutive frames. Window sizes of 13 and 29 frames are analyzed to study the trade-off between sensitivity and false-positive suppression. Real-world performance is then assessed on 15 unaltered full-length colonoscopy videos, again using temporally aggregated, frame-based evaluation. With a 13-frame window, the system achieves 89.3% sensitivity and an 8.3% false-positive rate, while a 29-frame window reduces false positives to 6.2% at a slightly lower sensitivity of 88.3% (Lee et al., 2020). However, the computational cost of the temporal post-processing is not given, and key detection parameters such as the underlying IoU criterion and confidence threshold for individual frame-level detections are not explicitly reported.

4.2 Modern Polyp Detection

Liu et al. propose a transformer-based polyp detection framework built on Deformable DETR. The authors design a data augmentation strategy that combines image normalization, geometric transformations (rotation, shearing, mirroring, and scaling), color perturbations, and additive noise. This augmentation pipeline expands the effective training set and explicitly targets the red background dominance and limited positional variability typical of endoscopic images (Liu et al., 2023). The baseline architecture is modified by inserting convolutional layers before each transformer encoder block to improve local feature extraction. In addition, the standard bounding-box regression loss is replaced with an improved loss formulation that includes distance and shape values for the IoU. Evaluation is performed on Kvasir, CVC-ClinicDB, and SUN, where only positive images are selected. Performance is reported using COCO-style object-detection metrics, including AP_{50} and AP_{75}. However, the paper does not specify the exact train–test split for this dataset. Across all datasets, the proposed augmented Deformable DETR variant consistently outperforms CNN-based models (Faster R-CNN, RetinaNet or YOLOv5) and the unmodified Deformable DETR, with the largest gains observed when extensive data augmentation is applied (Liu et al., 2023).

CHR-YOLO is single-stage detector is based on YOLOv8, and designed to improve contextual awareness and multi-scale feature fusion while reducing model complexity (Wan et al., 2024b). The method introduces a contextual receptive field enhancement module (CRFEM), a refined spatial pyramid pooling module (RSPPF), and a high-dimensional feature compensation structure to offset the removal of one detection head. Evaluation is done on Kvasir-SEG and a gastric polyp dataset with 700 images (Zhang et al., 2019) and only positive images are used for training and evaluation. Performance is reported using Precision, Recall, mAP_{50} and mAP_{50-95}, without specifying values for IoU or confidence. In addition to image-based experiments, the model is evaluated on the LDPolypVideo dataset to assess performance in video clips. The evaluation remains detection-centric and does not report confidence thresholds or information in dataset preparation.

Yoo et al. propose YOLOv5-TST, an hybrid architecture that integrates a transformer based token-sharing module (TST) into YOLOv5 to increase global context modeling while preserving real-time performance. The TST layer replaces part of the convolutional feature fusion in the neck and combines local convolutional features with global self-attention tokens. The approach is evaluated on the datasets SUN, KUMC, and Kvasir (and combinations) using precision, recall, and mAP (Yoo et al., 2024). The evaluation does not consider negative frames, or a validation set for training. Also, IoU or confidence are not given.

Wan et al. introduce a polyp detection model based on an enhanced YOLOv5 architecture for colonoscopy images, with the goal of improving detection accuracy over standard detectors. The authors incorporate a novel P-C3 module into both the backbone and neck of YOLOv5 to strengthen feature representation, and add a contextual feature augmentation (CFA) module at the base of the backbone to expand the receptive field and focus attention on polyp features. Experimental evaluation on 1200 polyp-only images shows that the proposed network achieves higher detection accuracy and especially better recall compared with other architectures like Faster R-CNN or YOLOv8 (Wan et al., 2024a). While the results indicate improved performance on still-image polyp detection, explicit evaluation settings like IoU thresholds or confidence are not reported.

4.3 Motivation

Even with lots of progress in automatic polyp detection, the practical relevance and comparability of reported results remain limited. A central factor across the literature is the way datasets are constructed and used. Many studies evaluate detectors exclusively on curated still images or short video clips, rather than on full-length colonoscopy procedures. Training and evaluation are often restricted to polyp-positive frames, while negative frames that dominate colonoscopy videos are heavily underrepresented or excluded entirely. In addition, dataset splits are frequently not stated or inconsistently defined. These choices simplify the experiment but do not reflect real-world scenarios. A second variability lies in the evaluation protocols and metrics. Studies differ in whether performance is assessed at the frame or detection level, and operating points such as confidence thresholds and IoU criteria vary in their sizes or are often not reported, despite their strong influence on sensitivity or recall. This thesis is motivated by the need for a systematic and reproducible evaluation of modern free-to-use real-time object detectors under realistic clinical conditions. The central goal is to compare representative detection architectures on full-length colonoscopy videos using fixed, patient-wise dataset splits, standardized evaluation metrics, and explicitly defined operating thresholds. By relying on out-of-the-box detector baselines and transparent experimental protocols, this work establishes a common reference point for assessing the practical applicability of enhanced polyp detection architectures.

5 Datasets

5.1 REAL-Colon

The primary dataset used in this thesis is REAL-Colon (Real-world multi-center Endoscopy Annotated video Library), a large-scale collection of full-length colonoscopy recordings designed to bridge the gap between academic benchmarks and clinical reality (Biffi et al., 2024).

The dataset has approximately 2.7 million frames extracted from 60 selected complete colonoscopy procedures. All data was sampled across four medical centers in the United States, Italy, Austria and Japan, thereby mitigating biases related to operator technique, procedural workflows and preparation standards (Biffi et al., 2024). Each video captures the entire examination from insertion to withdrawal without interruptions, preserving the natural variability of real-world colonoscopy such as rapid camera movement, fluctuating illumination, long periods without visible pathology or the extraction phase. All visible polyps were annotated by trained medical image specialists under gastroenterologist supervision. In total, the dataset contains 351 264 bounding-box annotations corresponding to 132 histologically verified colorectal polyps. Each frame in which a polyp appears is labeled. Notably, 87.6% of all frames contain no polyp, which reflects the true clinical distribution of colonoscopy video, where the vast majority of frames are negative (Biffi et al., 2024). This characteristic is essential for evaluating false positives and for assessing how detectors behave during long negative sequences. The authors further provide rich clinical metadata for every video, including patient age and sex, the bowel cleanliness measured using the Boston Bowel Preparation Scale (BBPS), endoscope manufacturer, original framerate and number of frames and lesions. At the polyp level, the dataset provides detailed histopathology (e.g., adenoma, sessile serrated lesion, hyperplastic), anatomical location, and size in millimetres.

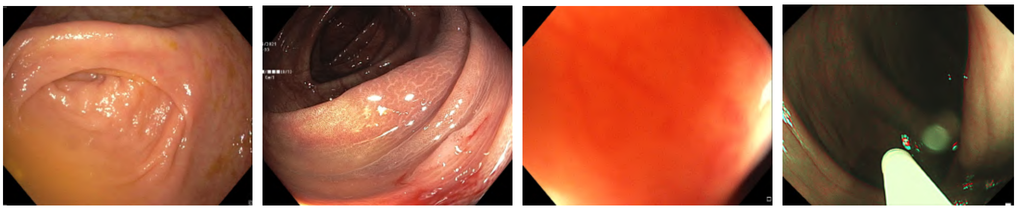


Figure 14: Representative frames extracted from the four distinct Institutions within the REAL-Colon dataset.

The polyp count and size distribution are shown in Figure 15 whereas the other values are given in the appendix 21. Procedures were recorded at native (1920 × 1080) resolution using Olympus and Fujifilm endoscope systems. The video stream was captured in YUV 4:2:2 with 10-bit color depth and then compressed using Apple ProRes to preserve image quality. Each video frame was then cropped to the endoscope's field of view to remove any on-screen identifiers and ensure full anonymization (Biffi et al., 2024). The resulting image sizes across splits and Institutions can

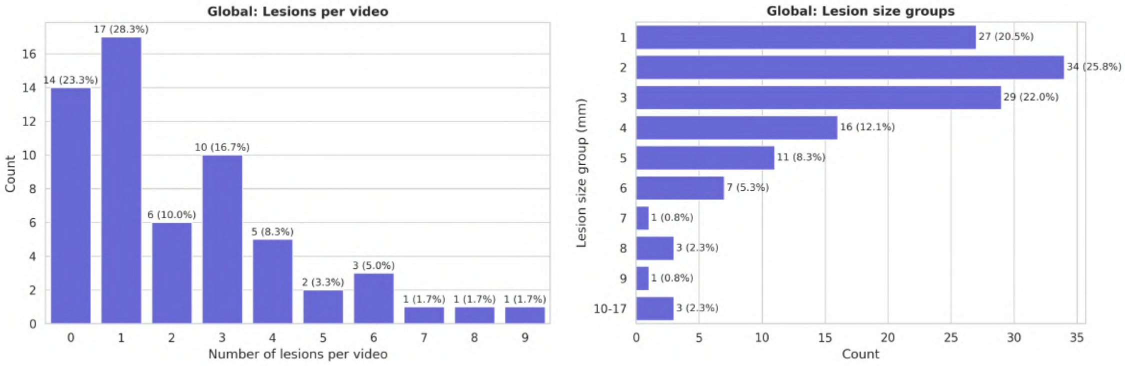


Figure 15: Dataset-level characteristics of REAL-Colon. Left: Distribution of polyp counts per procedure, showing that many recordings contain few or no polyps. Right: Distribution of the polyp sizes (maximum diameter in mm) across all annotated lesions.

be seen in Table 16 and 17 in the Appendix. For dataset construction, frames were extracted via ffmpeg and saved as high-quality JPEGs using `-qscale:v 1`, ensuring consistent and minimal compression artifacts across centers.

5.2 SUN-database

The Showa University and Nagoya University (SUN) database introduced by Misawa et al. (2021) is the second largest summary of polyp images containing complete histopathology information. The dataset was constructed from consecutive routine colonoscopies performed in Japan. All examinations were recorded with high-definition Olympus colonoscopes (CF-HQ290ZI and CF-H290ECI) and a dedicated video recorder (IMH-10), ensuring consistent image quality across cases. After applying exclusion criteria (inflammatory bowel disease, polyposis syndromes, nonepithelial lesions, low-quality or incomplete recordings, extraction proceudres and non-white-light imaging), the authors identified 1,635 candidate polyps, of which 100 were chosen for the final database, complemented by 13 negative colonoscopy videos without any polyps. This design aims to approximate an unbiased lesion spectrum while still providing a manageable benchmark set. Frame-level annotations and bounding boxes were generated by research assistants and refined by expert endoscopists, with a final quality assurance check performed by an independent external committee.

The resulting SUN-database consists of 152 560 frames in total, comprising 49 799 polyp-containing frames and 102 761 negative frames distributed across the 100 polyp videos and 13 negative videos. On the lesion level, the 100 polyps have a median size of 5 mm. Most lesions are protruded (66 protruded versus 34 flat). Histopathology is dominated by low-grade adenomas (82 lesions), but also includes hyperplastic polyps (7), sessile serrated lesions (4), traditional serrated adenomas (2), a small number of high-grade adenomas (4), and one invasive carcinoma.

5.3 PICCOLO

This is a publicly released dataset designed for training and evaluating supervised polyp detection, localisation, and segmentation models Sánchez-Peralta et al. (2020). It has 3 433 manually annotated frames from 76 lesions across 40 patients. In contrast to many prior datasets, PICCOLO includes both white-light (WL) and narrow-band imaging (NBI) frames and provides structured clinical metadata for every lesion (e.g., Paris classification, polyp size, and histopathology). Colonoscopy procedures were performed at Hospital Universitario Basurto (Bilbao, Spain) using Olympus CF-H190L and CF-HQ190L endoscopes. More than 145 000 frames were initially reviewed, of which 80 847 contained a visible polyp. To reduce temporal redundancy, one positive frame per second (every 25 frames) was sampled. After discarding uninformative frames (blur, noise, rapid transitions, or frames outside the patient), the final dataset includes 2 131 WL and 1 302 NBI images. Expert gastroenterologists produced manual segmentation masks for both the polyp and the void regions along the circular field of view, followed by post-processing to ensure mask completeness and consistency (Sánchez-Peralta et al., 2020). Images and anonymized patient data used in this study were provided by the PICCOLO database through the Basque Biobank (www.biobancovasco.bioef.eus).

6 Methods

This chapter explains the experimental methodology used to evaluate the applicability of real-time object detection algorithms for polyp detection in clinical colonoscopy videos. While previous chapters described the theoretical and architectural foundations of modern detectors, the focus here shifts to their practical application and clinical relevance.

6.1 Dataset Usage and Inclusion Criteria

6.1.1 Primary Dataset: REAL-Colon

By default, the dataset includes extended periods outside the patient where the operation room is visible. Since these frames hold no diagnostic relevance and risk biasing detectors towards non-endoscopic patterns, the dataset is restricted to *inside-patient* frames only. Frame ranges were selected manually for each recording. This filtering preserved 2,638,023 of 2,757,723 frames (95.7%), while excluding 119,700 frames (4.3%) containing non-endoscopic content. The frames *inside* remained untouched. The ranges of included images per video are in the Appendix in Table 18.

To avoid patient-level information leakage, the dataset was divided into training, validation, and test sets. Each recording was assigned entirely to one split, ensuring that evaluation reflects generalization to unseen patients rather than memorization of appearance patterns. Following the authors' documented split, the first 10 videos of each institution form the training set, videos 11 and 12 are assigned to validation, and the remaining videos 13 to 15 form the test set. The final training set contains 1,706,635 frames whereas validation and test have $359,303$ and $572,085$. Across the full dataset, 8,062 frames contain multiple annotated lesions. Under the adopted split, all such multi-lesion frames are moved in the training and validation sets, whereas the test split contains exactly one annotated lesion per lesion-positive frame.

6.1.2 Supplementary Datasets: SUN and PICCOLO

SUN Dataset This dataset is used for two purposes: as a cross-dataset evaluation source and for additional model training when combined with an independently constructed split. Unlike REAL-Colon, SUN consists of case-based video snippets rather than continuous full examinations, with a high proportion of no-polyp (negative) frames. In total the dataset contains 100 positive and 13 negative cases, with 49,136 images from positive cases and 109,553 images from negative cases (158,689 images overall).

Because the original dataset is intended for testing only and no precise split logic is given, a new partition was created for this work. Positive cases were divided sequentially, with the first 70 assigned to training, the following 10 to validation, and the remaining 20 to the test set. Negative cases were distributed using fixed counts per split (7 for training, 2 for validation and 4 for testing) and inserted at random positions between the positive sequences. This arrangement prevents clustering of negatives and produces a distribution that more closely resembles practical variation during deployment. The exact cases per split can be seen in the appendix 19.

PICCOLO Dataset This was included as a contrast to the video-based sources, since it contains only single-image samples. All original splits provided by the dataset authors were retained without modification. The material consists of paired polyp mask files (`Corrected.tif`), corresponding void masks (`Void.tif`), and the original `.png` images. In total, 2,203 samples belong to the training set, 897 to the validation set, and 333 to the test set, resulting in 3,433 images overall. The dataset was also used for pretraining for ablation studies.

Table 1: Dataset distribution including positive/negative frame counts per split and proportion of negative frames.

Dataset	Split	POS	NEG	Total	Neg (%)
REAL-Colon	Train	230,935	1,475,700	1,706,635	86.47
	Val	29,067	330,236	359,303	91.91
	Test	82,106	489,979	572,085	85.65
	All	**342,108**	**2,295,915**	**2,638,023**	**87.03**
SUN	Train	34,915	58,839	93,754	62.76
	Val	5,792	23,845	29,637	80.46
	Test	8,429	26,869	35,298	76.12
	All	**49,136**	**109,553**	**158,689**	**69.03**
PICCOLO	Train	2,183	20	2,203	0.91
	Val	869	28	897	3.12
	Test	332	1	333	0.30
	All	**3,384**	**49**	**3,433**	**1.43**

6.1.3 Annotation Conversion

All datasets used in this thesis were unified into COCO style annotation JSON for Detectron2 and normalized YOLO text files for Ultralytics-based models. The original annotation data (e.g., Pascal VOC XML for REAL-Colon, custom TXT for SUN, and mask files for PICCOLO) was first parsed to extract absolute pixel coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ and converted into COCO-style axis-aligned bounding boxes $(x_{\min}, y_{\min}, w, h)$. For REAL-Colon, every polyp was assigned a unique polyp identifier for further analysis. A COCO-compliant structure was generated containing image metadata, category definitions and annotation entries.

From these COCO files, YOLO labels were generated in a second conversion stage. For each bounding box, the coordinates were rescaled relative to image width W and height H, and expressed in center-normalized form (x_c, y_c, w, h), where:

$$x_c = \frac{x_{\min} + w/2}{W}, \qquad y_c = \frac{y_{\min} + h/2}{H}, \qquad w = \frac{w}{W}, \qquad h = \frac{h}{H},$$

and written to one `.txt` file per image containing the class index (0 for "lesion"). For images without annotated lesions, an empty label file was created, since Ultralytics-based YOLO training skips images that lack a corresponding annotation file. During conversion, filename patterns were formatted using zero-padding to ensure lexicographical frame ordering and mimic real video input for valuation when using batch 1. Final datasets were structured in the standard YOLO directory layout. The PICCOLO dataset uses masks and void files. The void file was used to remove external non-image artifacts and white borders. Bounding boxes are derived directly from the Polyp Mask. Lesions that are smaller than 20 pixels in area are removed, following the preparation in (Delaquintana-Aramendi et al., 2025).

6.2 Model Selection

This thesis evaluates a representative subset of object detection architectures that show both historical relevance and current technical progress in image analysis. Albuquerque et al. (2025) highlights Faster R–CNN, YOLO, and SSD as the most frequently adopted baselines for clinical object detection tasks, motivating the inclusion of Faster R–CNN as a classical two-stage reference. Since the original YOLO release, numerous architectural revisions have introduced stronger feature aggregation, improved bounding-box regression, and substantially faster inference throughput. In this context, YOLOv8 and YOLOv11 serve as one-stage detectors with strong real-time characteristics. The survey by Albuquerque et al. does not account for transformer-based detection, leaving this architectural direction underrepresented. RT-DETR is therefore selected, maintaining feasible training and inference cost in comparison to larger DETR variants. All detectors are used in their publicly released implementations without architectural modification.

6.2.1 Ultralytics Models (YOLOv8, YOLOv11, RT-DETR)

For all Ultralytics models, input processing and optimization are performed according to the framework's default settings, unless otherwise specified.

Image Loading and Preprocessing Input images are loaded using OpenCV (cv2) which reads them in BGR (Blue, Green, Red) channel order. Image dimensions are standardized using letterboxing to fit a square target resolution (e.g., 640×640), preserving the aspect ratio and padding the remainder with a uniform grayscale value of 114. The image data is represented as an 8-bit integer array (range [0, 255]) in (H, W, C) format, where H is height, W is width, and C is channels. The image is then prepared for PyTorch by transposing its dimensions to the (C, H, W) (channel-first) format. During this transposition, the BGR channel order is reversed to RGB. Final pixel normalization converts the data from 8-bit integers to floating-point values in the range [0.0, 1.0] via the scaling operation

$$\mathbf{I}_{\text{norm}} = \frac{\mathbf{I}_{\text{raw}}}{255.0}.$$

No further color standardization (i.e., mean subtraction or standard deviation division) is applied, as the defaults for `mean` and `std` used for the pretrained weights are $(0.0, 0.0, 0.0)$ and $(1.0, 1.0, 1.0)$, respectively.

Optimization and Training Configuration Optimizer selection is performed automatically by the Ultralytics framework. In all experiments conducted for this thesis, this internal logic consistently resulted in stochastic gradient descent (SGD) with an initial learning rate `lr0 = 0.01`, a momentum coefficient of `0.937`, and an L2-style regularization term `weight_decay = 0.0005`. Although the optimizer choice is decided by Ultralytics, it is consistent with the original SGD-based optimization used in the REAL-Colon SSD evaluation and in the Faster R–CNN reference configuration. The learning rate follows the standard scheduling strategy, where the final rate is defined as a fraction of the initial value using `lrf = 0.01`, resulting in a decay from `0.01` down to $0.01 \times 0.01 = 1 \times 10^{-4}$ over the training run. A short linear warm-up phase over the first three epochs stabilizes early optimization.

For the training phase, standard augmentations are applied to enhance model generalization and robustness. The default Ultralytics pipeline was used, which includes `mosaic` augmentation (combining four images into one batch element), random affine transformations (`scaling`, `translation`), `erasing` (removing random regions of the image), horizontal flipping, and HSV (Hue, Saturation, Value) augmentations with factors $(0.015, 0.7, 0.4)$.

The number of training `epochs` was set to 100 for all Ultralytics runs, combined with an early-stopping mechanism controlled by the `patience` parameter, which was chosen between 10 and 35 depending on dataset size and the resulting training time. Batch sizes were set with the (`autobatch`) function. It probes the maximum safe batch size for the given hardware at a specified input resolution. On the workstation with NVIDIA H100 NVL GPUs, this procedure yielded a stable batch size of 104 images per device for an image size of 640 × 640, so a total effective batch size of 208 was used. On the second workstation with an RTX A5000, the auto-tuning step resulted in a per-device batch size of 24 at the same resolution, leading to an effective batch size of 48 when using both GPUs.

Testing During testing, the YOLO's raw output predictions are filtered using NMS and an initial confidence threshold (`conf`) of 0.001 to further discard very low-confidence detections. RT-DETR produces a fixed set of object-query predictions per image (300 queries per default). While this design avoids post-hoc suppression, it leads to a large number of low-confidence predictions at evaluation time. When applied to large-scale video datasets, such as REAL-Colon, retaining all query outputs results in approximately 171 million predictions. To ensure tractable storage, predictions were post-processed by keeping only the top 100 predictions per image with a confidence score of at least 0.001.

6.2.2 Detectron2 (Faster R-CNN)

This framework is released by Facebook AI Research and provides an official PyTorch reimplementation of the original Caffe Faster R-CNN with Feature Pyramid Networks.

Image Loading and Normalization Images are read as RGB using the Python Imaging Library (`PIL`) and internally converted to BGR to match the convention used by the pretrained Detectron2 models. Pixel intensities remain in the $[0, 255]$ range and are normalized using channel-wise constants (mean = [103.53, 116.28, 123.675], std = [57.375, 57.12, 58.395]) that are fixed by the pretrained Faster R–CNN weights. All images are resized such that the shorter edge matches the configured minimum size while the longer edge is capped at a maximum value, preserving aspect ratio without letterboxing or distortion. The default Detectron2 pipeline applies minimal geometric transforms with `RandomHorizontalFlip`, which flips the image and corresponding ground-truth bounding boxes with a probability of 0.5. Empty annotation files are retained to ensure that negative frames contribute to the learning, and data loading is parallelized with 16 workers to maintain high I/O throughput.

Optimization and Training Configuration The base model configuration is derived from the default `faster_rcnn_R_50_FPN_3x.yaml`, modified only to allow a single lesion class and the custom COCO-style REAL-Colon dataset definitions. Optimization follows Detectron2's default stochastic gradient descent with momentum (0.9) and weight decay (0.0001). Detectron2 does not provide automatic batch tuning, and memory requirements for Faster R–CNN are significantly higher, making YOLO-derived batch configurations infeasible. Default batch sizes in Detectron2 typically lie between four and eight images per GPU, which would lengthen training beyond reasonable limits for the scale of this study. Instead, batch sizes were empirically determined through manual testing to avoid out-of-memory failures while maintaining acceptable throughput. A batch size of 48 was identified as a stable compromise resulting in a `total batch` of 96 used across two GPUs (for a maximum edge size of 640). Detectron uses iterations instead of epochs, thus the maximum iteration count is set to 267000, representing approximately 15 epochs for 1,706,635 training images ($\lceil 1{,}706{,}635/96 \rceil \approx 17{,}777$ iterations per epoch). A multi-step decay schedule reduces the learning rate at `STEPS = (160000, 213500)`, corresponding to roughly 60% and 80% of total training. Automatic mixed precision is enabled to reduce memory requirements and accelerate training without sacrificing accuracy. Training is executed through a customized `ThesisTrainer` subclass of Detectron2's `DefaultTrainer`, which registers the REAL-Colon dataset splits and attaches a `COCOEvaluator` for periodic validation and final test evaluation. The final predictions were also filtered using NMS and a confidence value set to 0.001 to match the Ultralytics setup.

6.3 System Environment and Hardware Setup

All experiments were executed on two dedicated workstation systems, ensuring consistent and reproducible computational resources for training and evaluation. Workstation 1 served as the primary execution environment for all REAL-Colon experiments and all final evaluations reported, while Workstation 2 was used exclusively for training models on supplementary datasets.

Hardware Configuration

- **Workstation 1 (Primary Execution)**: Equipped with two Intel Xeon Gold 5416S CPUs (64 hardware threads total) and 503 GiB of system memory. It has two NVIDIA H100 NVL GPUs, each with 95,830 MiB of VRAM. Data was stored on a local NVMe SSD to ensure high I/O throughput during training.
- **Workstation 2 (Supplementary Training)**: It uses a single Intel Xeon W-2265 processor (24 logical threads), has 125 GiB of system memory and includes two NVIDIA RTX A5000 GPUs, each with 24,564 MiB of VRAM.

Software Environment

Both systems run Ubuntu 24.04.3 LTS. The NVIDIA driver version was 575.57.08 on Workstation 1 and 580.95.05 on Workstation 2, with CUDA version 12.9 driving the primary execution system. The software environments were managed via Conda:

- **Ultralytics (Shared)**: Python 3.10.19, `ultralytics` 8.3.232, and `PyTorch` 2.6.0+cu124. This environment was used for all YOLOv8, YOLOv11, and RT-DETR training and evaluation runs across both workstations.
- **Detectron2 (Workstation 1 Only)**: Python 3.10.19, `pycocotools` 2.0.10, `Detectron2` 0.6 and `PyTorch` 2.5.1+cu121.

6.4 Experimental Design

The experiments are structured to answer two central questions: (i) how well basic modern object detectors perform on full-procedure colonoscopy data under realistic conditions, and (ii) whether this performance is compatible with real-time clinical deployment. To address these questions, several complementary experiment types are defined: a core architectural comparison on REAL-Colon, ablation studies using YOLOv11 and cross-dataset transfer evaluations. Training configurations for Ultralytics and Detectron2 follow the framework-specific setups detailed in Section 6.2, and all runs are executed on the hardware described in Section 6.3.

6.4.1 Baseline Performance on REAL-Colon

The main experiment compares four object detection architectures under a unified training and evaluation protocol on the REAL-Colon dataset. All models are trained, validated, and tested on the same splits. The comparison isolates architectural behavior under realistic, *out-of-the-box* conditions instead of tuning each detector independently. To assess result stability and reduce the influence of random initialization, each experiment is repeated with three different random seeds (0, 42, and 123).

All experiments use a single foreground class, *lesion*, while all negative frames are kept as background, following the best performing variant of the original REAL-Colon study. Pretrained weights are used as provided by the respective frameworks, consistent with the authors' baseline relying on the NVIDIA SSD detector with ImageNet pretraining. Input images are resized to a resolution of 640×640 pixels for all models. This resolution corresponds to the default configuration of the Ultralytics framework. Although Detectron2 has a default input size of 1333×800, this setting would substantially increase the already long training times on REAL-Colon and was therefore deemed infeasible. Prior experiments indicated no measurable loss in detection performance when reducing the input size to 640×640, motivating the use of a unified resolution across all architectures. Training durations, either in epochs or iterations, follow the configurations defined in Section 6.2. For Ultralytics-based models, early stopping is enabled to mitigate unnecessary overfitting and excessive training time. After training, the best-performing model checkpoint determined on the validation set is selected for each run (based on the $\mathrm{mAP}_{50:95}$). These checkpoints are then evaluated on the held-out test set.

Evaluation is performed at multiple levels to capture complementary aspects of detector behavior, using the metrics defined in Section 3.5. Detection accuracy is assessed using comparable COCO-style metrics computed with `pycocotools`, including mean Average Precision and Average Recall at multiple overlap thresholds. The results for the remaining experiments depend strongly on the choice of the spatial overlap. Following the evaluation protocol of Troya et al. (2024), a permissive IoU threshold of $\mathrm{IoU} > 0$ is adopted, such that any spatial overlap between a prediction and a ground-truth lesion is considered sufficient. This choice is better aligned with the objective of assessing whether a lesion is detected at all, rather than how precisely it is localized. At the same time, predictions that do not overlap with any annotated lesion are still counted as false positives. Detection operating characteristics are then analyzed using Alternative FROC (AFROC) curves, which plot sensitivity against the false positive fraction (FPF) as the detection confidence threshold is varied. Frame-level alert behavior is further evaluated by collapsing detections within each frame into a binary decision and computing sensitivity-, specificity-, and precision-based metrics at selected IoU and confidence operating points. This aligns with the evaluations done on other full procedure literature (Troya et al., 2024; Biffi et al., 2024). In addition, lesion-level event statistics are computed by aggregating detections across all frames belonging to the same annotated polyp. Detection consistency shows the fraction of frames in which a given polyp is correctly identified, while early detection is characterized by the temporal distance between the first ground-truth appearance and the first frame in which it is detected. Some lesions disappear for over 1,500 frames after a brief initial appearance. To prevent these artifacts from inflating the results, lesions with a gap of more than 50 frames within the first 250 frames of appearance were removed from the mean latency calculation. This affected 2 lesions (12 and 14 in Figure 18). Detection within fixed temporal window of 1, 3, and 5 seconds is additionally evaluated, where a lesion is considered detected only if it is detected in at least 15 frames within that interval, corresponding to a detection at 30 FPS. Finally, real-time suitability is assessed by measuring per-frame inference latency and the resulting effective frame rate (FPS) at batch size 1. For Faster R-CNN, timings are taken from Detectron2, which reports inference time and a combined end-to-end iteration time that includes evaluation overhead. For Ultralytics models, runtimes are taken from per-image speed breakdown including preprocessing, inference, and postprocessing. One iteration corresponds to one processed image for Faster R-CNN.

6.4.2 Ablation Experiments

To understand the sensitivity of the results to design and training choices, some ablation experiments on YOLOv11 are done. YOLOv11 is selected because it represents the most recent Ultralytics one-stage design considered in this thesis, offers multiple capacity variants under a consistent implementation, and has higher inference speed compared to two-stage or transformer-based alternatives. All experiments are reported for a single random seed (42). Several ablation axes are considered:

Input Resolution To evaluate the trade-off between spatial detail and computational efficiency, YOLOv11 was trained using multiple square input resolutions, ranging from lower resolutions (e.g. 224×224) to higher resolutions (e.g. 640×640 and 1024×1024). In addition, all detector families were trained and evaluated at an input resolution of 300×300. The resulting performance is compared to the baseline reported by (Biffi et al., 2024). In their study, the authors evaluated the dataset using the NVIDIA SSD implementation trained at an input resolution of 300×300. Their model was initialized with ImageNet-pretrained weights and trained for 65 epochs using standard SSD preprocessing, including resizing to the target resolution, random cropping, photometric augmentations (brightness, contrast, saturation, and hue), horizontal flipping, and ImageNet mean–variance normalization. The reported training configuration used a batch size of 96. Replicating the SSD training schedule exactly for all models would require substantially longer training times than feasible within the scope of this thesis (approximately eight to twelve days per model for 65 epochs). Therefore, the SSD results are included as published. All models evaluated in this work follow the training and preprocessing pipeline described in the previous sections. A single technical adjustment was required for the YOLO-based detectors as the model needs square input dimensions divisible by the model stride (32). The

300 × 300 configuration was internally mapped to 320 × 320 via letterboxing and padding, preserving the effective field of view.

Capacity, Sampling, and Optimization The robustness is further evaluated across model capacity, training negative ratios, and optimizer selection. For Model Capacity, variants of different scales (S, M, and L) are trained on the same split at an input resolution of 640 × 640 to determine how detection performance and inference latency scale with network size. Next, using the same resolution, the effect of class imbalance in REAL-Colon is analyzed by varying the negative sampling ratio between 1:0.15, 1:0.5, and a balanced 1:1 ratio of positive to negative frames per training epoch. This setup differs from the total-frame-based sampling strategy described in (Biffi et al., 2024), where negative ratios are defined relative to the full dataset negative pool rather than as a specific positive-to-negative balance. The ablation evaluates whether a high frequency of background frames is required for model selectivity or whether lower negative sampling rates suffice to maintain stable optimization. Finally, Optimizer Selection compares the default SGD against adaptive methods (Adam and AdamW) at a reduced resolution of 224 × 224 for faster iterative testing.

6.4.3 Cross-Dataset Transfer

The objective is to evaluate whether detection models trained on full-procedure recordings (REAL-Colon) maintain applicabile when applied to data from different institutions, and annotation protocols (SUN, PICCOLO) without further adaptation. First, to establish a baseline for comparison, all four architectures are trained and evaluated using dataset-internal splits for the SUN and PICCOLO datasets. These runs provide a reference of the achievable performance under in-domain conditions. Next, the transferability of the models, trained exclusively on the REAL-Colon dataset, is assessed through evaluation on the test splits of SUN and PICCOLO without any fine-tuning. For these cross-dataset evaluations, only frame-level metrics can be computed, as the other datasets do not provide lesion identifiers for lesion-level aggregation. Performance is therefore evaluated using $\text{mAP}_{50:95}$ and mAP_{50}, as well as true positive and false positive rates where applicable. For PICCOLO, evaluation based on the FPR is not meaningful because the test split contains only a single negative frame (out of 333), restricting achievable values to either 0% or 100%.

7 Results

This chapter presents the experimental results for real-time lesion detection in colonoscopy videos and evaluates them with respect to detection performance and practical deployability. Following the experimental design in Section 6.4, the evaluation is organized into three parts: (i) a base architectural comparison on the full-procedure REAL-Colon dataset under a unified training protocol, (ii) targeted ablation studies on YOLOv11 to analyze the influence of selected design and training parameters, and (iii) cross-dataset experiments to assess generalization to SUN and PICCOLO without adaptation.

7.1 Baseline Performance on REAL-Colon

The evaluation proceeds from generic detection accuracy, over detection-level operating characteristics, to frame-level alert behavior and finally lesion-level and temporal performance to present the increasing clinical abstraction from bounding boxes to actionable assistance. Results are reported as mean ± standard deviation over three random seeds.

7.1.1 Overall Detection Accuracy

Table 2 reports standard detection metrics where Average Precision is reported at multiple IoU thresholds, together with Average Recall under increasing limits on the number of predictions per image.

Table 2: Detection performance on REAL-Colon.

	Faster R-CNN	YOLOv8	YOLOv11	RT-DETR
mAP_{50}	0.324 ± 0.032	0.413 ± 0.002	0.384 ± 0.009	0.488 ± 0.019
mAP_{75}	0.216 ± 0.023	0.299 ± 0.010	0.263 ± 0.005	0.359 ± 0.009
$\text{mAP}_{50:95}$	0.196 ± 0.021	0.272 ± 0.005	0.245 ± 0.005	0.327 ± 0.009
AR_{1}	0.295 ± 0.014	0.376 ± 0.003	0.357 ± 0.005	0.415 ± 0.011
AR_{10}	0.387 ± 0.030	0.466 ± 0.009	0.453 ± 0.012	0.585 ± 0.007
AR_{100}	0.394 ± 0.035	0.471 ± 0.012	0.457 ± 0.013	0.672 ± 0.008

Absolute $\text{mAP}_{50:95}$ values range from 0.196 (Faster R-CNN) to 0.327 (RT-DETR), indicating that consistent, high-precision localization remains challenging for all models on full-procedure colonoscopy data. Even at the more permissive mAP_{50} level, performance remains moderate, with RT-DETR achieving the highest value (0.488) and the remaining architectures clustering at lower accuracy. Across all models, Average Precision decreases as the IoU threshold is increased from 0.5 to 0.75 and further to the $\text{mAP}_{50:95}$ aggregate, reflecting the increasing difficulty of precise bounding-box alignment under stricter overlap requirements. The relative drop between mAP$_{50}$ and mAP$_{75}$ is most pronounced for Faster R-CNN, where mAP$_{75}$ falls to 0.216, indicating that while coarse lesion localization is frequently achieved, precise spatial alignment is less reliable. Among the one-stage detectors, YOLOv8 consistently outperforms YOLOv11 across all reported Average Precision metrics. The difference is relatively small, with YOLOv8 exceeding YOLOv11 by 0.027 in mAP$_{50:95}$ but it still indicates that newer models do not automatically perform better. Faster R-CNN trails all other models across IoU thresholds with the lowest Average Precision values overall. Average recall generally increases as the prediction limit is raised from one to ten and further to one hundred. However, recall for the CNN-based models saturates early, with only marginal gains between AR$_{10}$ and AR$_{100}$. In contrast, RT-DETR has a substantially larger recall increase by 0.087 between AR$_{10}$ and AR$_{100}$ and reaching an AR$_{100}$ of 0.672, suggesting that its query-based, NMS-free design preserves a richer set of candidate detections, even if many are assigned lower confidence scores.

Table 3 further breaks down Average Precision by object size according to the COCO definition (small: $\leq 32^2$, medium: 32^2–96^2, large: $\geq 96^2$ pixels).

Table 3: Average Precision by object size.

	Faster R-CNN	YOLOv8	YOLOv11	RT-DETR
AP$_{\text{small}}$	0.000	0.000	0.000	0.002 ± 0.002
AP$_{\text{medium}}$	0.012 ± 0.006	0.049 ± 0.009	0.054 ± 0.005	0.044 ± 0.014
AP$_{\text{large}}$	0.207 ± 0.021	0.284 ± 0.005	0.255 ± 0.005	0.343 ± 0.010

Detection performance varies a lot across object sizes. At a fixed input resolution of 640 × 640, the COCO size categories correspond to objects covering at most 0.25% (small), 0.25–2.25% (medium), and more than 2.25% (large) of the image area (Lin et al., 2014). Average Precision for small objects is effectively zero for all evaluated models, with RT-DETR reaching only 0.002. Performance for medium-sized objects remains limited, with AP$_{\text{medium}}$ ranging from 0.012 (Faster R-CNN) to 0.054 (YOLOv11). Even for RT-DETR, AP$_{\text{medium}}$ reaches only 0.044, showing that improved global reasoning does not resolve this limitation. Detection performance is therefore mainly done by objects classified as large. In this range, RT-DETR achieves the best AP$_{\text{large}}$ with 0.343. Consequently, most successful detections occur in frames where lesions occupy a relatively large fraction of the image, so when the camera is zoomed in, while detection of smaller and more distant lesions remains challenging across all architectures.

7.1.2 Detection Operating Characteristics

While COCO-style metrics summarize localization accuracy at fixed overlap criteria, they do not describe how detection sensitivity and false localizations trade off as the confidence threshold is varied. To analyze this operating behavior, detection performance is evaluated using the alternative free-response receiver operating characteristic (AFROC) curves. Figure 16 shows the detection sensitivity as a function of the false positive fraction across the full range of confidence thresholds. The AFROC curves have a steep initial increase, indicating that a significant portion of bounding boxes can be recovered with a low false-positive rate (FPF < 0.1). RT-DETR demonstrates the highest performance with a distinct sensitivity advantage across the entire FPF range. The YOLO-based detectors form a clustered intermediate tier, with YOLOv8 consistently yielding slightly higher sensitivity than YOLOv11 at equivalent FPF levels. This is consistent with the small performance differences observed in detection-level metrics. Faster R-CNN shows lower sensitivity at comparable false positive fractions. Even when higher false positive rates are tolerated, its sensitivity remains below that of the other detectors.

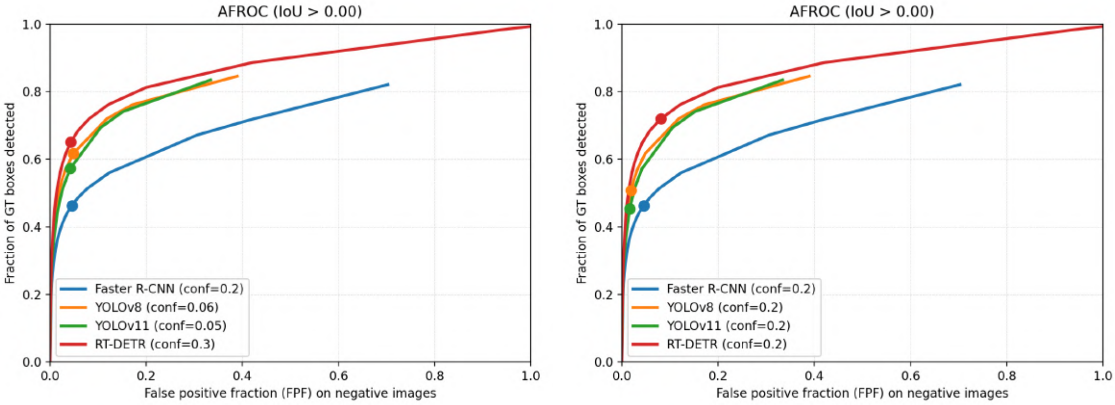
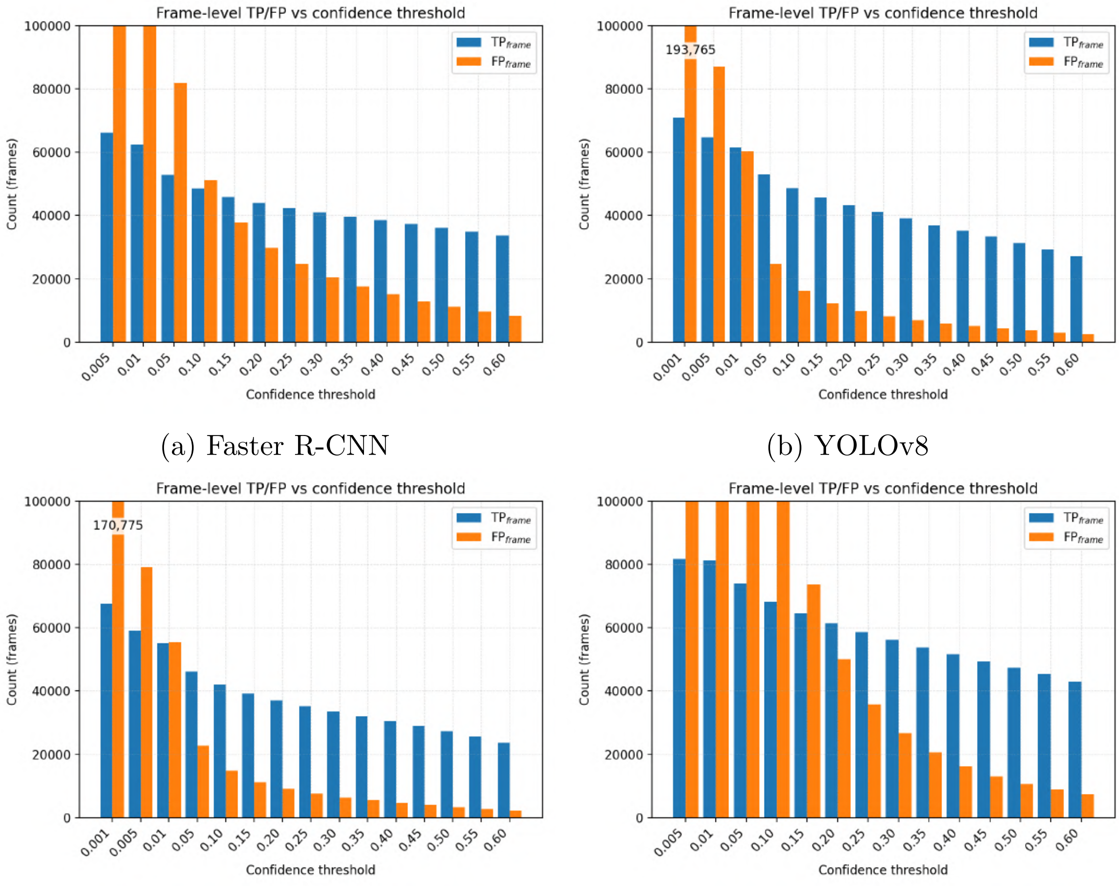


Figure 16: AFROC curves on REAL-Colon

The operating markers on the curves visualize the performance at the specific confidence thresholds used and explained in the subsequent frame-level analysis. These markers illustrate the threshold-dependent trade-off: lowering the confidence threshold shifts the operating point to the right along the curve, increasing sensitivity at the cost of additional false positives, while increasing the threshold moves it to the left. Notably, the YOLO markers lie on a steep gradient of the curve, implying that minor threshold adjustments can lead to large shifts in false-positive behavior. In contrast, RT-DETR maintains the target FPF at a much higher confidence (0.30), indicating a more robust separation between true detections and background noise. AFROC is prioritized for this comparison as it provides a bounded false-positive metric [0,1]. Standard FROC curves are provided in Figure 22 (Appendix) but are less suitable for combined plotting due to large ranges of False Positives Per Image (FPPI).

7.1.3 Frame-Level Alert Performance

To complement detection-level metrics, frame-level performance is evaluated by collapsing all detections within a frame into a single binary decision.



(a) Faster R-CNN

(b) YOLOv8

(c) YOLOv11

(d) RT-DETR

Figure 17: TP and FP rate on REAL-Colon at IoU > 0 for varying confidence thresholds (seed 42).

Confidence Threshold Selection This score directly controls the trade-off between true and false alerts and is therefore strongly model dependent. A fixed confidence value can lead to substantially different true- and false-positive rates across detection architectures. Figure 17 visualizes this trade-off for all four architectures at IoU > 0. Two confidence operating points are reported throughout this experiment. First, a fixed confidence threshold of 0.2 is used for all models to align with the evaluation protocol of Troya et al. (2024) and to provide a common reference. Second, a model-specific operating point is selected to get comparable frame-level false-positive rates across architectures. The threshold is chosen such that the resulting false-positive rate is approximately 4–5% for each model, matching the FPR given in the original REAL-Colon paper (Biffi et al., 2024). For this, a single confidence value is chosen per architecture and applied uniformly across all three random seeds. Based on this criterion, the confidence thresholds are set to 0.06 for YOLOv8, 0.05 for YOLOv11, and 0.30 for RT-DETR. Faster R-CNN naturally reaches the targeted FPR at the baseline confidence of 0.20, so no adjustment is applied. The analysis is not intended to optimize detector predictions, but to define representative and comparable operating points.

Frame-Level Metrics at Selected Operating Points In this experiment, sensitivity (TPR) is the primary indicator of whether a system successfully raises an alert on frames that contain at least one visible lesion. A frame is counted as a TP as soon as the lesion is detected, so sensitivity reflects the probability that a lesion-present frame triggers an alert. Table 4 summarizes the results.

Table 4: Frame-level performance using multiple confidence thresholds.

Metric	**Faster R-CNN**	**YOLOv8**	**YOLOv11**	**RT-DETR**
Confidence 0.20				
Sensitivity / TPR	0.463 ± 0.066	0.508 ± 0.022	0.453 ± 0.004	0.720 ± 0.041
specificity	0.954 ± 0.017	0.980 ± 0.002	0.983 ± 0.001	0.918 ± 0.030
FPR	0.046 ± 0.017	0.020 ± 0.002	0.017 ± 0.001	0.082 ± 0.030
Precision	0.637 ± 0.086	0.811 ± 0.014	0.816 ± 0.011	0.606 ± 0.085
F_1 score	0.532 ± 0.047	0.624 ± 0.016	0.582 ± 0.005	0.654 ± 0.030
F_2 score	0.488 ± 0.057	0.549 ± 0.020	0.497 ± 0.004	0.691 ± 0.011
Confidence Faster R-CNN: 0.20, YOLOv8: 0.06 YOLOv11: 0.05, RT-DETR: 0.30				
sensitivity / TPR	0.463 ± 0.066	0.605 ± 0.030	0.573 ± 0.009	0.651 ± 0.049
specificity	0.954 ± 0.017	0.956 ± 0.006	0.957 ± 0.003	0.957 ± 0.017
FPR	0.046 ± 0.017	0.044 ± 0.006	0.043 ± 0.003	0.043 ± 0.017
Precision	0.637 ± 0.086	0.698 ± 0.023	0.693 ± 0.020	0.723 ± 0.070
F_1 score	0.532 ± 0.047	0.648 ± 0.015	0.627 ± 0.013	0.681 ± 0.001
F_2 score	0.488 ± 0.057	0.621 ± 0.024	0.594 ± 0.011	0.662 ± 0.031

Note: An extended analysis with confidence thresholds selected per seed (to match ≈ 4% FPR per run) is provided in Appendix 21. Under this setting, frame-level metrics for RT-DETR and Faster R-CNN vary less across seeds.

At the fixed reference threshold of 0.20, the architectures show fundamentally different default behaviors. RT-DETR prioritizes recall, achieving the highest sensitivity (0.720) but at the cost of significant background noise (FPR ≈ 0.08). In contrast, the YOLO-based models operate conservatively, maintaining high specificity (> 0.98) and precision (> 0.81). This results in missed detections, with sensitivity being lower than those of RT-DETR. This is also seen in the F scores: while the balanced F_1 scores are relatively close, RT-DETR achieves better scores for F_2 (0.691 vs. 0.549 for YOLOv8).

Variance across random seeds differs notably between architectures under fixed-threshold evaluation. Precision shows higher variability for RT-DETR and Faster R-CNN (≈ 0.085) than for the YOLO-based detectors (< 0.014). Since threshold-independent detection metrics (mAP) remain comparatively stable across seeds for all models, this frame-level difference indicates that the calibration of confidence scores varies between training runs for the transformer and R-CNN architectures, whereas YOLO models have more consistent score distributions.

When confidence thresholds are adjusted to achieve comparable false positive rates across models, sensitivity values converge while remaining distinct. Under these normalized operating points, RT-DETR retains the highest sensitivity (0.651), followed by YOLOv8 (0.605) and YOLOv11 (0.573). This adjustment reduces the performance gap but shifts the precision, which decreases for the YOLO-based detectors (due to lowering thresholds) and increases for RT-DETR (due to raising thresholds). Correspondingly, F_1 and F_2 scores improve notably for the YOLO-based models. However, even after this optimization, RT-DETR maintains the highest F_2 score (0.662). The relative ordering of the models is consistent with the AFROC analysis, confirming that frame-level alert behavior mimics the fundamental sensitivity–false-positive trade-offs observed at the detection level.

7.1.4 Lesion-Level Consistency and Early Detection

The lesion-level analysis evaluates whether a lesion is detected at all, how persistently it is detected across its visible duration, and how quickly the first detection occurs after the lesion appears. Table 5 reports the number of lesions detected within 1, 3, and 5-second windows, requiring a minimum of 15 frames of detection per interval. Across all operating points, every model successfully identifies all 21 lesions at least once (100% lesion-level sensitivity). However, this binary criterion is insufficient to characterize clinical utility. More informative are the consistency and early-detection metrics, which better reflect whether a lesion is reliably highlighted during its visible duration. At the fixed confidence threshold of 0.20, RT-DETR demonstrates superior stability and speed. It maintains detections for at least 50% of the visible frames for nearly all lesions (mean 19.7), whereas the YOLO family and Faster R-CNN average only 12.7 and 11.7, respectively. RT-DETR also acts significantly faster: it detects 7.7 lesions within the first second of appearance, with an average latency of just 12.5 frames ($\approx$ 0.5 seconds). In contrast, the other architectures have a substantial delay, requiring over 60 frames ($\approx$ 2.5 seconds) on average.

Adjusting thresholds to normalize the false-positive rate shows that for the YOLO models, lowering the threshold allows them to capture earlier, lower-confidence features. This effectively halves the latency (e.g., YOLOv8 improves from 82.1 to 46.8 frames) and improves consistency scores. In contrast, raising the threshold for RT-DETR filters out the earliest, most uncertain predictions. This results in a “latency penalty”, where the mean time to first detection increases from 12.5 to 38.3 frames, and the number of lesions detected within the first second drops from 7.7 to 4.7.

Table 5: Lesion-level consistency and latency on REAL-Colon. Values represent the mean number of unique lesions detected (n=21) and the average latency to first detection (in frames)

	Faster R-CNN	YOLOv8	YOLOv11	RT-DETR
Fixed confidence threshold 0.20				
Lesions detected (≥ 1 match)	21	21	21	21
Lesions detected (≥ 25% frames)	19	19.3	18.7	21
Lesions detected (≥ 50% frames)	11.7	12.7	12.7	19.7
Detected within 1 s	1	1.7	1	7.7
Detected within 3 s	7	5.3	5.3	12.0
Detected within 5 s	11.0	8.0	7.7	15.3
Latency first frame	63.9 ± 42.5	82.1 ± 30.6	103.0 ± 37.6	12.5 ± 7.2
Confidence Faster R-CNN: 0.20, YOLOv8: 0.06 YOLOv11: 0.05, RT-DETR: 0.30				
Lesions detected (≥ 1 match)	21	21	21	21
Lesions detected (≥ 25% frames)	19	20.7	21	21
Lesions detected (≥ 50% frames)	11.7	16.0	15.7	17.3
Detected within 1 s	1	3.7	3	4.7
Detected within 3 s	7	8.7	7.3	10
Detected within 5 s	11	12.3	10.7	12.7
Latency first frame	63.9 ± 42.5	46.8 ± 26.3	57.3 ± 26.3	38.3 ± 17.3

Despite this shift, the adapted RT-DETR remains the fastest and most temporally consistent architecture, followed closely by the adapted YOLOv8. Faster R-CNN consistently shows the highest latency (63.9 frames) and lowest temporal stability.

To further illustrate lesion-level behavior beyond aggregate metrics, Figure 18 visualizes the fraction of detected frames for each individual lesion across random seeds at the selected operating points. For each lesion, the mean number of detected and missed frames is shown, together with the minimum and maximum detection counts observed across seeds. This visualization highlights lesion-specific differences in detection persistence and variability that are not captured by global averages alone.

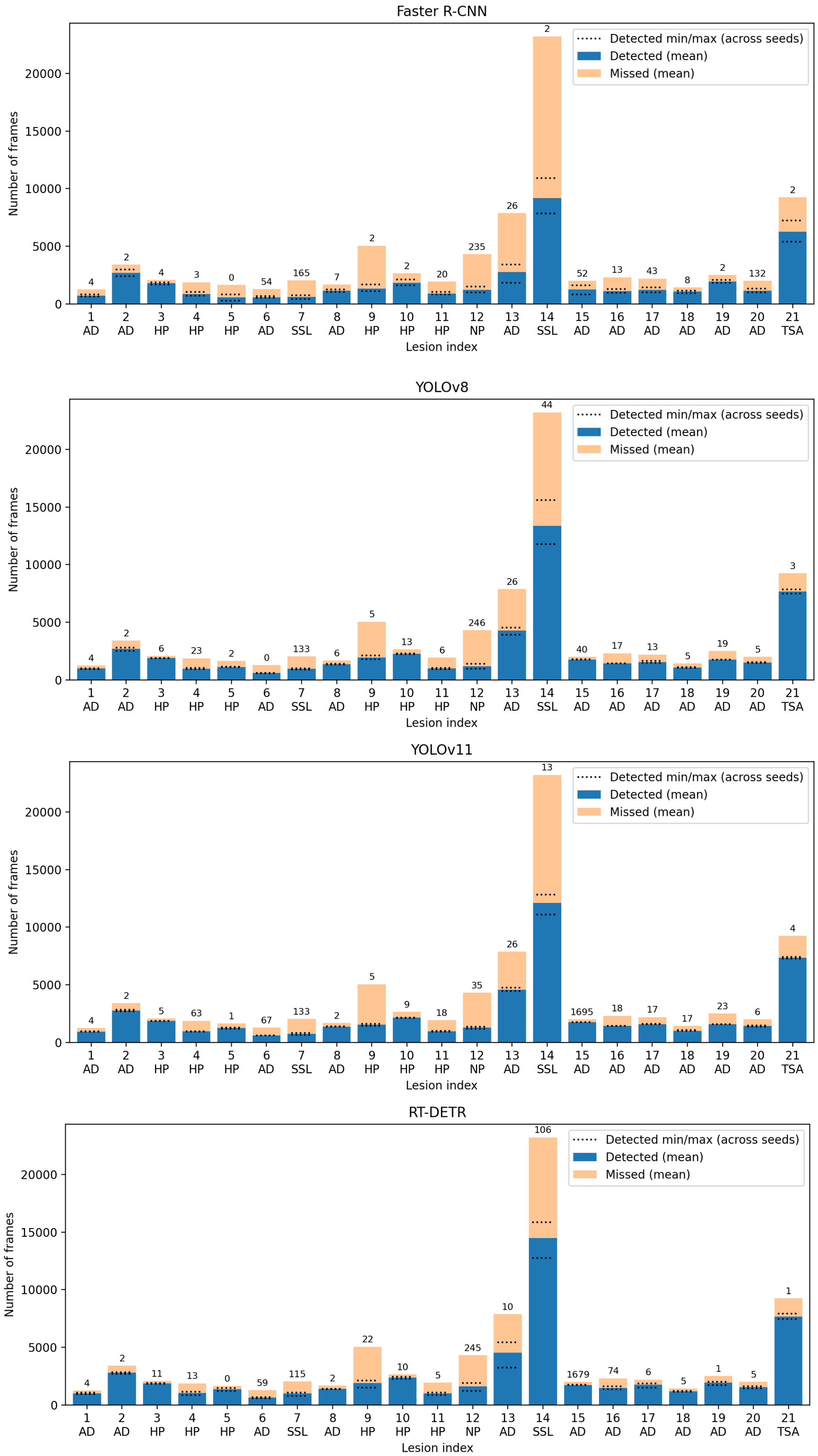


Figure 18: Fraction of frames detected for each lesion using specific confidence values. Number on top of the bars is the first detected frame

7.1.5 Runtime and Real-Time Suitability

Finally, Table 6 reports component-wise runtime measurements. Two measures are reported: FPS_{inf} is computed from inference time only, while $\text{FPS}_{\text{total}}$ shows end-to-end latency based on all reported components.

Table 6: Runtime characteristics on REAL-Colon at batch size 1 on a single NVIDIA H100 GPU

Detectron2

Model	Inf [ms]	Eval [ms]	Total [ms]	FPS_{inf}	$\text{FPS}_{\text{total}}$
Faster R-CNN	25.07 ± 2.72	0.20 ± 0.00	26.70 ± 2.52	39.9	37.5

Ultralytics

Model	Pre [ms]	Inf [ms]	Post [ms]	Total [ms]	FPS_{inf}	$\text{FPS}_{\text{total}}$
YOLOv8	0.57 ± 0.06	4.53 ± 0.38	0.40 ± 0.00	5.50 ± 0.44	220.7	181.8
YOLOv11	0.50 ± 0.00	5.20 ± 0.17	0.40 ± 0.00	6.10 ± 0.17	192.3	163.9
RT-DETR	0.50 ± 0.00	18.87 ± 0.29	0.23 ± 0.06	19.60 ± 0.35	53.0	51.0

The YOLO-based architectures achieve the highest throughput among all evaluated models. YOLOv8 attains the lowest total latency (5.50 ± 0.44 ms per frame), corresponding to an end-to-end throughput of 181.8 FPS, followed by YOLOv11 with 163.9 FPS. These rates exceed typical video frame rates (25–60 Hz) by a factor of 3 to 7. RT-DETR has higher latency than the YOLO-based detectors but maintains an end-to-end throughput of 51.0 FPS. This places RT-DETR above commonly used real-time frame rates, yet with a smaller margin compared to the YOLO family. Faster R-CNN shows the highest total latency among the evaluated models (26.7 ± 2.52 ms), resulting in an end-to-end throughput of 37.5 FPS. While this exceeds 30 FPS on the tested hardware, it provides less margin relative to the one-stage detectors.

7.2 Ablation Studies on REAL-Colon

To isolate the influence of specific design and training parameters, a series of ablation experiments were done using YOLOv11. Evaluation is performed at model-specific confidence thresholds chosen to get comparable frame-level false-positive rates with the IoU > 0 criterion.

Image Resolution The trade-off between spatial detail and computational cost is evaluated by training YOLOv11-M at three distinct resolutions. Table 7 summarizes the results. Increasing the input resolution from 224×224 to 640×640 yields a large performance gain, improving mAP_{50} by 0.095 and frame-level sensitivity by 0.14 to 0.584. However, scaling further to 1024×1024 offers lower returns in accuracy

Table 7: Effect of input resolution.

Resolution	conf	$\mathrm{mAP}_{50:95}$	mAP_{50}	TPR	FPR	$\mathrm{FPS}_{\mathrm{inf}}$	$\mathrm{FPS}_{\mathrm{total}}$
224 × 224	0.01	0.184	0.289	0.444	0.044	212.9	183.4
640 × 640	0.05	0.246	0.384	0.584	0.043	192.3	163.9
1024 × 1024	0.03	0.255	0.4	0.576	0.044	184.3	143.8

(mAP_{50} +0.016) while increasing computational load, identifying 640 × 640 as the efficient operating point.

A secondary comparison was made at 300 × 300 resolution to align with the original REAL-Colon study (Biffi et al., 2024). Frame-level TPR and FPR depend on the specific IoU and confidence thresholds, which are not specified in the original author setup, and are therefore reported as an observational comparison only.

Table 8: Comparison at 300 × 300 resolution following the original author setup on the full REAL-Colon dataset.

Model	conf	$\mathrm{mAP}_{50:95}$	mAP_{50}	TPR	FPR
Faster R-CNN	0.2	0.203	0.33	0.497	0.043
YOLOv8-M	0.013	0.221	0.342	0.524	0.048
YOLOv11-M	0.015	0.194	0.297	0.504	0.046
RT-DETR	0.25	0.29	0.432	0.634	0.05
SSD-300 (author setup)		0.216	0.338	0.505	0.054

RT-DETR maintains the highest results in this constrained regime, followed by YOLOv8 and the original SSD-300 baseline. YOLOv11-M trails slightly behind YOLOv8-M at this resolution, mirroring the trend observed in the main experiments.

Capacity, Sampling, and Optimization Table 9 analyzes the robustness of the architecture across model capacity, negative sampling strategies, and optimizer selection.

Table 9: Effect of various hyperparameters

Model	conf	$\mathrm{mAP}_{50:95}$	mAP_{50}	TPR	FPR
YOLOv11-S	0.04	0.221	0.353	0.570	0.045
YOLOv11-M	0.05	0.246	0.384	0.584	0.043
YOLOv11-L	0.007	0.276	0.420	0.628	0.044
Negative ratio 0.15	0.27	0.209	0.328	0.482	0.048
Negative ratio 0.50	0.27	0.155	0.254	0.45	0.047
Negative ratio 1.00	0.1	0.253	0.401	0.602	0.048
Adam	0.005	0.181	0.285	0.474	0.05
AdamW	0.07	0.182	0.285	0.5	0.048
SGD	0.05	0.184	0.289	0.444	0.044

Performance scales positively with network depth and width. The Large variant (YOLOv11-L) achieves the highest detection sensitivity (0.628), while the S-model shows a notable drop in mAP_{50} to 0.353, suggesting that sufficient parameter capacity is required to capture polyp morphology effectively. Yet larger models come with an increase in the overall training time (from 15min per epoch for S to 1.5h to model L). Regarding dataset composition, the balanced setup (Ratio 1:1) yields the highest detection performance (mAP_{50} 0.401). Reducing the proportion of negative samples degrades performance. Notably, the more imbalanced configuration ($1 : 0.15$ and $1 : 0.50$) forces the model to adopt a much higher confidence threshold (0.27) to control false positives, resulting in lower sensitivity. This indicates that a sufficient amount of background examples is essential for learning model selectivity. Finally, at lower resolutions (224×224), the architecture demonstrates robustness to the choice of optimizer, with Adam, AdamW, and SGD achieving nearly identical mAP values (0.285 to 0.289).

7.3 Cross-Dataset Evaluation

This section evaluates dataset generalization across multiple colonoscopy datasets. First, in-domain reference performance is established for the SUN and PICCOLO datasets using dataset-internal train–test splits. Second, cross-dataset transfer is assessed by evaluating YOLOv11-M models trained on a single source dataset on unseen target datasets without further fine-tuning. Tables 10 and 11 report results when models are trained and evaluated on the same dataset. On the clip-based SUN dataset, all architectures achieve high and closely clustered detection performance, with mAP_{50} values between 0.689 and 0.738 and comparable frame-level sensitivities (TPR ≈ 0.73). Differences between models are minor. RT-DETR attains slightly lower mAP values and the highest FPR, but remains competitive overall.

Table 10: Evaluation on the SUN dataset.

Model	conf	Train → Test	$\mathrm{mAP}_{50:95}$	mAP_{50}	TPR	FPR
Faster R-CNN	0.07	SUN → SUN	0.408	0.738	0.739	0.044
YOLOv8	0.07	SUN → SUN	0.396	0.694	0.731	0.046
YOLOv11	0.1	SUN → SUN	0.411	0.724	0.735	0.047
RT-DETR	0.12	SUN → SUN	0.389	0.689	0.725	0.05

Table 11: Evaluation on the PICCOLO dataset.

Model	Train → Test	$\mathrm{mAP}_{50:95}$	mAP_{50}
Faster R-CNN	Piccolo → Piccolo	0.531	0.743
YOLOv8	Piccolo → Piccolo	0.487	0.673
YOLOv11	Piccolo → Piccolo	0.56	0.77
RT-DETR	Piccolo → Piccolo	0.186	0.321

On the image-based PICCOLO dataset, a clearer separation between architectures is observed. Faster R-CNN, YOLOv8, and YOLOv11 maintain high accuracy, with mAP_{50} values ranging from 0.673 to 0.770, while RT-DETR performs substantially worse, reaching only 0.321 mAP_{50}. This suggests that the transformer-based architecture struggles significantly with the specific image characteristics or smaller scale of the PICCOLO dataset compared to its CNN-based counterparts. Compared to SUN, larger variations in $\text{mAP}_{50:95}$ are observed across models, reflecting differences in object scale and annotation characteristics between datasets.

Table 12: Cross-dataset evaluation (Direct Transfer). Comparison across architectures.

Model	Train → Test	Conf	$\text{mAP}_{50:95}$	mAP_{50}	TPR	FPR
Faster R-CNN	SUN → REAL-Colon	0.2	0.131	0.220	0.29	0.043
YOLOv8	SUN → REAL-Colon	0.25	0.105	0.177	0.261	0.045
YOLOv11	SUN → REAL-Colon	0.040	0.097	0.164	0.240	0.040
RT-DETR	SUN → REAL-Colon	0.25	0.132	0.225	0.310	0.040
Faster R-CNN	Piccolo → REAL-Colon	0.99	0.08	0.154	0.271	0.06
YOLOv8	Piccolo → REAL-Colon	0.8	0.024	0.051	0.175	0.046
YOLOv11	Piccolo → REAL-Colon	0.7	0.031	0.059	0.134	0.041
RT-DETR	Piccolo → REAL-Colon	0.8	0.005	0.012	0.007	0.003
Faster R-CNN	REAL-Colon → SUN	0.12	0.251	0.503	0.514	0.049
YOLOv8	REAL-Colon → SUN	0.006	0.365	0.714	0.767	0.048
YOLOv11	REAL-Colon → SUN	0.06	0.361	0.705	0.765	0.043
RT-DETR	REAL-Colon → SUN	0.08	0.379	0.717	0.728	0.043
Faster R-CNN	REAL-Colon → Piccolo	-	0.379	0.645	-	-
YOLOv8	REAL-Colon → Piccolo	-	0.314	0.558	-	-
YOLOv11	REAL-Colon → Piccolo	-	0.304	0.554	-	-
RT-DETR	REAL-Colon → Piccolo	-	0.336	0.602	-	-

Table 12 details the direct transfer performance across all architectures. The results reveal a consistent asymmetric transfer effect independent of the model family. Models trained on the curated datasets (SUN, PICCOLO) have a big performance decline when applied to the full-procedure REAL-Colon dataset. For the SUN-trained models, mAP_{50} values drop to between 0.16 and 0.22. Transfer from the image-based PICCOLO dataset is even less effective, with mAP_{50} values falling below 0.16 for all architectures. Notably, Faster R-CNN demonstrates slightly higher robustness in this direction compared to the YOLO family and RT-DETR. In the reverse direction, models trained on REAL-Colon generalize effectively to external domains. On the SUN dataset, YOLOv8, YOLOv11, and RT-DETR achieve mAP_{50} scores exceeding 0.7, closely matching the in-domain baselines established in Table 10. Faster R-CNN shows weaker transfer in this specific case ($\text{mAP}_{50} = 0.503$). On the PICCOLO dataset, generalization remains strong across all models, with Faster R-CNN achieving the highest transfer accuracy ($\text{mAP}_{50} = 0.645$), followed by RT-DETR (0.602).

8 Discussion

This discussion interprets the experimental findings with respect to the central motivation of this thesis: establishing a transparent and realistic baseline for real-time polyp detection under full-procedure colonoscopy conditions. The results are discussed in terms of their combined implications for detector behavior, robustness, and clinical applicability.

8.1 Architectural Trade-offs: Bias and Calibration

Sensitivity vs. Selectivity (Baseline Behavior) The comparative analysis reveals a fundamental difference in how CNN-based (YOLO) and Transformer-based (RT-DETR) architectures approach the polyp detection task. RT-DETR demonstrates a strong recall bias, consistently achieving the highest frame-level sensitivity (0.720) at the baseline confidence threshold of 0.20. This suggests that the global attention mechanism effectively captures lesion context even in challenging frames, such as those with blur or partial occlusion. However, this sensitivity comes at the cost of reduced specificity, resulting in the highest False Positive Rate ($\approx 8\%$), often triggered by non-pathological structures such as folds, bubbles, or specular highlights. In contrast, the YOLO family has a more conservative operating behavior. Both YOLOv8 and YOLOv11 maintain a very high specificity (> 0.98) and precision (> 0.81), but at the cost of lower sensitivity (0.508 and 0.453).

From a clinical screening perspective, where missed polyps (false negatives) generally carry greater risk than additional false alarms, RT-DETR's dominance in the recall-weighted F_2 score (0.691 compared to 0.549 for YOLOv8) highlights its suitability for safety-critical applications, provided that the elevated false-positive rate can be controlled through appropriate threshold selection.

Confidence Stability Beyond absolute performance, the experiments reveal differences in confidence stability across architectures. While detection-level metrics such as mAP remain more stable across random seeds for all models, as they do not rely on such thresholds, larger variation is observed at the frame level under fixed-threshold evaluation, especially for Faster R-CNN and RT-DETR ($\approx \pm 0.085$). In contrast, the YOLO-based models show consistently lower variance in frame-level precision (< 0.014), indicating more stable confidence score distributions. When confidence thresholds are adjusted per seed to achieve comparable false-positive rates, performance differences across architectures narrow (see Table 21). RT-DETR retains the highest recall-oriented scores, including the F_2 metric. This implies that while RT-DETR is the most powerful detector, it requires dynamic threshold tuning (per-model or per-seed) to guarantee a specific operating point, whereas YOLO models offer a more "plug-and-play" deployment where fixed thresholds yield predictable behavior.

8.2 Clinical Applicability and Benchmarking

To place these findings into a broader context, Table 13 reproduces metrics reported for several commercial CADe systems and the alternative EndoMind by Troya et al. (2024). The table is included to illustrate the range of sensitivities and false-positive rates reported for colonoscopy CADe systems under frame-level evaluation at IoU > 0 and a fixed confidence threshold of 0.2.

Table 13: Results of commercial CADe systems reported in (Troya et al., 2024).

Metric	**GI Genius v1**	**GI Genius v2**	**EndoAID A**	**EndoAID B**	**EndoMind**
Sensitivity / TPR	0.5063	0.6785	0.6560	0.5295	0.6022
Specificity	0.9694	0.9577	0.9719	0.9930	0.9589
FPR	0.0275	0.0380	0.0252	0.0063	0.0369
Precision	0.5966	0.5701	0.6658	0.8720	0.5499
F_1 score	0.5915	0.6391	0.6943	0.7177	0.5960
First detection (in ms)	1510	607	659	1316	1083

The open-source detectors evaluated here achieve comparable sensitivity levels (0.60-0.72) but at slightly higher false positive rates (0.045 vs. 0.03 commercial). Regarding latency, the commercial systems report first detection times between 600 and 1500 ms. In comparison, the raw latency of the evaluated open-source models is slightly higher, ranging from 1.2 s (RT-DETR) to 2.1 s (Faster R-CNN) at the specific confidence.

While the evaluation was made on a different dataset and under a distinct pipeline, this comparison suggests that modern, openly available detection architectures can reach frame-level performance similar to commercial systems, even without some post-processing or data-specific tuning that is done in the study. At the same time, differences in datasets, image selection criteria, annotation protocols, and evaluation process limit the interpretability of such comparisons and highlight the shortcomings of frame-level metrics as the only indicator of clinical performance.

Temporal Dynamics and Latency Artifacts To assess clinical reliability beyond raw speed, detection consistency is evaluated using a strict persistence criterion ($\geq$ 15 frames) to distinguish robust alerts from flickering (Table 5). Immediate, reliable detection within the first second is rare. Under this setting, reliable detection within the first second of lesion appearance is uncommon across all evaluated architectures. At specific operating points, the YOLO-based models detect approximately 3 to 4 out of 21 lesions within this window, while RT-DETR detects fewer than 5 lesions. Even within the first 3 seconds, fewer than half of the lesions are reliably detected by most models. This indicates that while modern detectors eventually identify all lesions, they require a longer period to get enough visual evidence, potentially delaying clinically actionable warnings during rapid camera movements.

Real-Time Feasibility and Hardware All models met the real-time threshold (> 30 FPS) on an NVIDIA H100. However, throughput of the YOLO models (e.g. > 180 FPS for YOLOv8) provide a massive safety margin that allows deployment on edge devices (e.g., directly on the endoscope processor) or leaves computational headroom for concurrent tasks (e.g., depth estimation, pathology classification, post-negative filtering). From a practical perspective, the large gap between inference-only and end-to-end FPS is small for all models, indicating that preprocessing, postprocessing, and evaluation overhead contribute only marginally to overall latency under the tested conditions.

8.3 Data-Centric Insights and Generalization

The experiments demonstrate that dataset composition is as critical as architectural choice for robust polyp detection.

Universal Generalization Asymmetry The experiments reveal a consistent generalization asymmetry across all evaluated model families. Models trained on the full-procedure videos of REAL-Colon transfer robustly to external benchmarks. For instance, all but Faster R-CNN match their respective in-domain baselines when evaluated on the SUN dataset, indicating that the feature representations learned from full-procedure data are sufficiently diverse to encompass the visual distributions of curated clip-based datasets. In contrast, the transfer in the reverse direction fails universally. Regardless of the architecture, models trained on curated clips or static images degrade to unusable performance levels when exposed to the REAL-Colon test set. While Faster R-CNN shows slightly higher robustness in this direction, it still fails to bridge the domain gap. This confirms that curated datasets represent a limited subset of the endoscopic domain, lacking the required diversity of lighting, angles, and negative samples found in continuous procedures.

Training Efficiency and Redundancy Ablation studies further highlighted that a balanced (1:1) negative sampling ratio yielded detection performance ($\text{mAP}_{50} = 0.401$) competitive with, and even slightly exceeding, the baseline trained on the full set of negative frames ($\text{mAP}_{50} = 0.384$). Which contrasts the findings of the original authors (Biffi et al., 2024). This suggests that the vast majority of background frames in full procedures do not contribute unique information to the decision boundary. For future development, this implies that training efficiency can be drastically improved by aggressive subsampling of background frames without compromising model selectivity.

8.4 Qualitative Error Analysis

Lesion Consistency The variance observed in the lesion consistency metric (Table 5, $\geq 50\%$ frames) is driven by specific challenging lesions rather than uniform stochastic noise. As visualized in Figure 18, while robust lesions (e.g., indices 2 and 3) are detected consistently across all models, a subset of “borderline” lesions (e.g., 4, 11, or 14) are detected in only about 50% of their visible frames. For these cases, the detection density lies close to the threshold. Thus, minor stochastic differences in training initialization determine whether a specific lesion effectively passes or fails this criterion. The Histology Split analysis further reveals that Sessile Serrated Lesions (SSL, indices 7 and 14) and the Traditional Serrated Adenoma (TSA, index 21) are exclusively present in the test set, meaning the models are performing *zero-shot* detection on these subtypes. The TSA (Lesion 21) is detected with high consistency across all architectures, suggesting its visual features share strong similarities with the common Adenomas (AD) seen during training. In contrast, the SSLs (Lesions 7 and 14) have much higher miss rates and variance, often falling into the borderline category. This suggests that SSLs have distinct visual characteristics that do not fully transfer from an AD/HP-only training distribution. Visual samples of these zero-shot targets are provided in Appendix in Figure 23.

Visualizing Failure Modes Figure 19 presents a qualitative comparison of model predictions (red) versus ground truth (green) to illustrate some behavioral patterns. In ideal conditions with clearly visible polyps, all architectures produce accurate detections (Row 1). However, specific instabilities emerge in more complex scenarios. For relatively large lesions (Row 2), YOLOv8 and RT-DETR occasionally generate fragmented predictions, placing multiple bounding boxes on a single object. While the lesion is successfully retrieved, these duplicate detections penalize precision metrics. Sensitivity differences are highlighted in Rows 3 and 6: Faster R-CNN fails to detect a small lesion distracted by specular reflections (Row 3), and both Faster R-CNN and YOLOv11 miss a lesion due to specific endoscopic lighting (Row 4). Severe visual degradation defeats all evaluated models. Lesions heavily masked by specular highlights (Row 5) or significant motion blur (Row 6) result in universal false negatives. Finally, Row 7 shows the trade-off between sensitivity and semantic understanding: while Faster R-CNN correctly ignores an operating instrument, the YOLO variants and RT-DETR misclassify the instrument as a lesion.

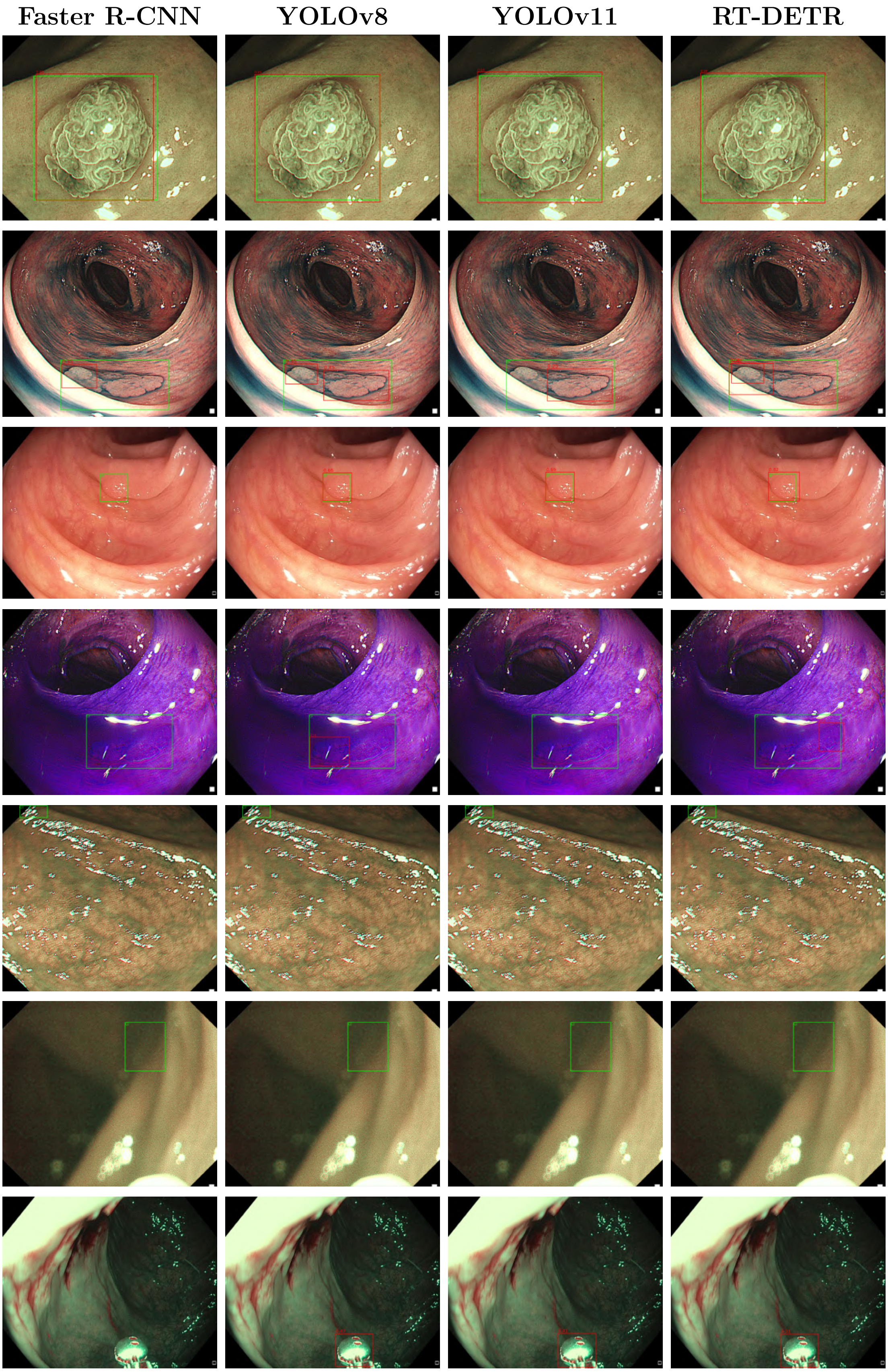


Figure 19: Comparison of detection results on REAL-Colon. Ground-truth annotations are shown in green, predictions in red.

9 Limitations and Future Work

Limitations Several limitations of this study should be considered when interpreting the reported results. First, regarding experimental design, confidence thresholds were selected either globally or adjusted to match false-positive rates, rather than being optimized for each specific architecture. Similarly, hyperparameter tuning was intentionally kept consistent based on the used framework to ensure comparability. While this isolates the "out-of-the-box" behavior of the architectures, it may disadvantage models that rely on extensive, architecture-specific tuning to reach peak performance. Second, the temporal analysis identified discontinuities in the ground truth annotations. To address this, lesions were excluded if an annotation gap exceeding 50 frames occurred within the first 250 frames of their initial appearance. However, shorter interruptions were not explicitly filtered and may persist in the evaluation set. Finally, evaluations were performed on a high-end NVIDIA H100 GPU. The absolute inference speeds on this hardware do not necessarily reflect the constraints of embedded clinical systems. Moreover, as all experiments were conducted offline on pre-recorded video, the study cannot assess the "human-in-the-loop" dynamics, such as how endoscopists react to false positives or whether algorithmic alerts actually improve polyp retrieval rates in live procedures.

Future Work From a modeling perspective, broader architectural coverage and more extensive hyperparameter optimization could be explored. Future benchmarks should evaluate emerging architectures that increase the accuracy-latency trade-off. RF-DETR (Receptive Field-based Detection Transformer) (Robinson et al., 2025) has recently achieved SOTA results (> 0.60 $\text{mAP}_{50:95}$ on COCO), outperforming current YOLO and RT-DETR variants by using Neural Architecture Search (NAS) to optimize receptive fields. Additionally, newer iterations of the RT-DETR family, such as RT-DETRv4 (Liao et al., 2025), which uses vision foundation model distillation, can be assessed to determine if their architectural refinements translate to improved robustness in the endoscopic domain. The integration of Foundation Models in general shows promising results. Recent work by Delaquintana-Aramendi et al. (2025) demonstrated that while zero-shot application of models like YOLO-World failed (< 0.001 AP), fine-tuning foundation models such as Grounding DINO (Liu et al., 2024) yielded best detection performance on the PICCOLO dataset ($\text{mAP}_{50:95}0.805$). However, this validation was limited to curated image datasets (PICCOLO). Given the finding that training on REAL-Colon results in better results, a next step would be to train these foundation models directly on full-procedure video data. Finally, future work should address deployment-oriented considerations. Evaluating performance on a wider range of hardware platforms, including resource-constrained edge devices, would improve the practical relevance of runtime analyses. Integration with real-time endoscopy systems and user-facing interfaces would be necessary steps toward translating algorithmic performance into clinical use.

10 Conclusion

This thesis addresses the gap between academic polyp-detection benchmarks and the complex conditions of clinical colonoscopy. Prior work often relies on curated images or short clips, loosely defined operating points, and differently reported metrics, limiting comparability and clinical relevance. To counter this, a transparent and reproducible evaluation framework was established, integrating detection-, frame-, and lesion-level metrics on full-procedure colonoscopy videos.

The results show that modern general-purpose object detectors approach the performance of specialized medical CADe systems under realistic conditions. A clear architectural trade-off emerges. RT-DETR achieves the highest sensitivity and temporal consistency, favoring safety-critical screening, but at higher computational cost and reduced confidence stability. In contrast, YOLO-based models provide highly efficient and predictable behavior, combining very high throughput with strong specificity, making them well suited for resource-constrained deployment.

Beyond architecture, data composition proves to be the dominant factor for generalization. Models trained on full-procedure video generalize robustly to external datasets, whereas models trained on curated clips fail under procedural variability. Further, detector performance is highly dependent on the chosen evaluation abstraction. Models that appear competitive under detection-level metrics can behave very differently once predictions are aggregated temporally or constrained by fixed false-positive budgets. As a result, conclusions drawn from still-image or frame-isolated evaluations do not necessarily translate to full-procedure deployment. Future research and benchmarking efforts must therefore explicitly define operating points, aggregation rules, and dataset composition to enable meaningful comparison and to better reflect clinical requirements.

However, various challenges remain for clinical translation. The analysis of temporal dynamics shows that all evaluated architectures struggle to reliably detect lesions across several frames during the initial seconds of appearance, particularly when polyps are small or distant. This latency, often masked by aggregate metrics, points to a limitation in current single-frame detection paradigms.

In summary, this thesis shows that progress toward robust, clinically applicable polyp detection depends not only on architectural advances, but more fundamentally on realistic data, transparent evaluation protocols, and metrics that capture temporal behavior under true procedural conditions.

A Appendix

A.1 Evolution of YOLO Architectures

Table 15 summarizes the key architectural and methodological advancements introduced across major YOLO versions, highlighting how design choices have progressively shifted toward improved efficiency, training stability, and deployment flexibility.

Table 14 indicates that recent progress is focused more in reducing parameter counts and computational complexity while maintaining comparable detection accuracy, enabling deployment on smaller or resource-constrained devices. Given the rapid release cycle of approximately two YOLO versions per year, older models remain relevant when robustness or deployment constraints are more important than minor performance gains.

Table 14: Evolution of YOLO architectures and performance on the COCO (2017) benchmark.

Model	mAP$_{50:95}$	mAP$_{50}$	Params [M]	FLOPs [G]
YOLOv2 (COCO 2015)	21.6	44.0	–	–
YOLOv3 (COCO 2015)	33.0	57.9	–	–
YOLOv4	38.9	60.7	12.0	–
YOLOv5m	45.4	64.1	21.2	49.0
YOLOXm	46.4	65.4	25.3	73.8
YOLOv6m	49.5	66.8	34.3	82.2
YOLOv7m	51.4	69.7	36.9	104.7
YOLOv8m	50.2	67.2	25.9	78.9
YOLOv9m	51.4	68.1	20.0	76.3
YOLOv10m	51.1	68.1	15.4	59.1
YOLOv11m	51.5	68.5	20.1	68.0

Table 15: Key architectural and methodological advancements across successive YOLO versions.

Model	Main Advancements
YOLOv2	(Redmon and Farhadi, 2016) introduces the DarkNet-19 CNN backbone, batch normalization, anchor boxes via K-means clustering, multi-scale training and ImageNet-pretrained classifier for improved feature extraction and convergence.
YOLOv3	(Redmon and Farhadi, 2018) uses the DarkNet-53 backbone with residual connections creating three output layers of different spatial resolutions, Feature Pyramid Network-style multi-scale prediction, multi-label classification with cross-entropy loss and improved small-object detection.
YOLOv4	(Bochkovskiy et al., 2020) adapts the CSPDarknet53 backbone based on CSPNet, improving gradient flow by splitting feature maps into parallel learning and propagation paths (Sapkota et al., 2025), integrates Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet) for cross-level feature fusion, and introduces mosaic data augmentation combining four images into different contexts to enable more stable training.
YOLOv5	(Jocher, 2020) transitions from the Darknet framework to PyTorch, improving modularity and deployment, and introduces SPPF (fast SPP) for efficient multi-scale context aggregation.
YOLOX	(Ge et al., 2021) introduces anchor-free detection, decoupled classification and regression heads, and simplified label assignment to improve generalization and reduce anchor-related heuristics.
YOLOv6	(Li et al., 2023) is designed for edge and mobile deployment and introduces the EfficientRep backbone derived from RepVGG, enabling parallel computation and efficient structural reparameterization.
YOLOv7	(Wang et al., 2022) proposes the Efficient Layer Aggregation Network (ELAN) and Efficient-ELAN to improve gradient flow under model scaling, together with a unified scaling strategy across depth, width, and resolution.
YOLOv8	(Jocher et al., 2023) employs C2f (Cross-Stage Partial with two convolutions) backbone modules, optimizes bounding-box regression using CIoU loss, and adopts a simplified detection head with binary cross-entropy classification.
YOLOv9	(Wang and Liao, 2024) uses Programmable Gradient Information (PGI) as an additional supervision mechanism to mitigate information loss during deep feature transformations, and proposes the GELAN backbone combining ELAN and CSPNet principles to improve parameter efficiency.
YOLOv10	(Wang et al., 2024) removes non-maximum suppression at inference via a consistent dual-label assignment strategy that combines one-to-many supervision during training with one-to-one matching at inference time.

A.2 Dataset Supplementary Information

Table 16: Total and per Split frame resolution distribution in REAL-Colon.

Resolution	Train	Val	Test	Total
1352×1080	1,133,606	183,814	344,318	1,661,738
1350×1080	247,807	0	56,272	304,079
1248×959	115,788	72,938	45,347	234,073
1162×1007	147,058	0	36,869	183,927
1244×1080	74,584	0	54,167	128,751
1246×1080	24,491	42,420	0	66,911
1164×1034	0	0	54,674	54,674
1164×1010	0	48,470	0	48,470
1158×1008	29,108	0	0	29,108
1160×1052	0	23,527	0	23,527
1158×1024	22,465	0	0	22,465

Table 17: Frame resolution per Institution (001-004) in REAL-Colon.

Resolution	001	002	003	004	Total
1352×1080	504,894	0	1,156,844	0	1,661,738
1350×1080	0	304,079	0	0	304,079
1248×959	0	234,073	0	0	234,073
1162×1007	0	0	0	183,927	183,927
1244×1080	0	0	0	128,751	128,751
1246×1080	42,420	0	0	24,491	66,911
1164×1034	0	0	0	54,674	54,674
1164×1010	0	0	0	48,470	48,470
1158×1008	0	0	0	29,108	29,108
1160×1052	0	0	0	23,527	23,527
1158×1024	0	0	0	22,465	22,465

Table 18: REAL-Colon inside-patient frame ranges per video. Only frames with image IDs within the listed interval were retained

Video	**Range**	**Video**	**Range**	**Video**	**Range**
1_1	920–45137	1_2	1620–25974	1_3	1967–39372
1_4	850–45385	1_5	360–28544	1_6	786–48217
1_7	1050–37920	1_8	1165–34891	1_9	2031–39884
1_10	2075–46215	1_11	380–23390	1_12	530–40842
1_13	1996–22417	1_14	347–28711	1_15	910–30831
2_1	1–25421	2_2	414–24916	2_3	1–47191
2_4	1–30537	2_5	70–40342	2_6	670–52180
2_7	989–49622	2_8	568–32638	2_9	1447–28957
2_10	312–25540	2_11	420–37747	2_12	1–33739
2_13	330–22515	2_14	785–55308	2_15	583–20772
3_1	415–51456	3_2	990–49874	3_3	12240–105451
3_4	6342–75787	3_5	4980–122040	3_6	4478–102060
3_7	6932–58000	3_8	1702–50387	3_9	1071–58101
3_10	452–55736	3_11	1617–62313	3_12	1041–96306
3_13	773–73379	3_14	2427–75752	3_15	2757–110858
4_1	699–25264	4_2	687–34245	4_3	1491–33222
4_4	583–33629	4_5	298–21781	4_6	326–40377
4_7	593–28828	4_8	1145–22604	4_9	1600–30189
4_10	925–23927	4_11	786–48100	4_12	1338–22971
4_13	1300–54128	4_14	459–36698	4_15	327–53698

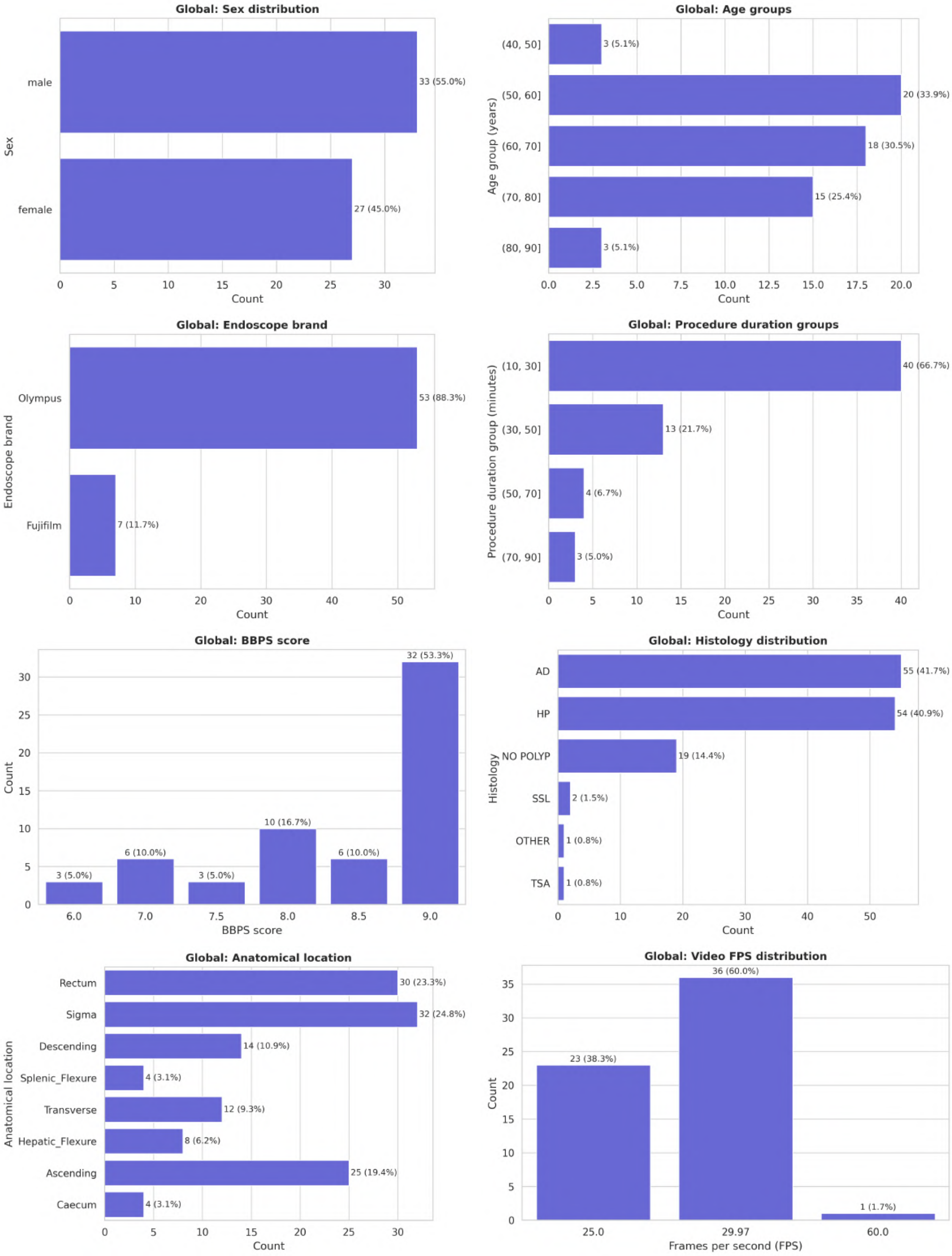


Figure 20: Lesion-level and preparation characteristics in the REAL-Colon dataset summarized.

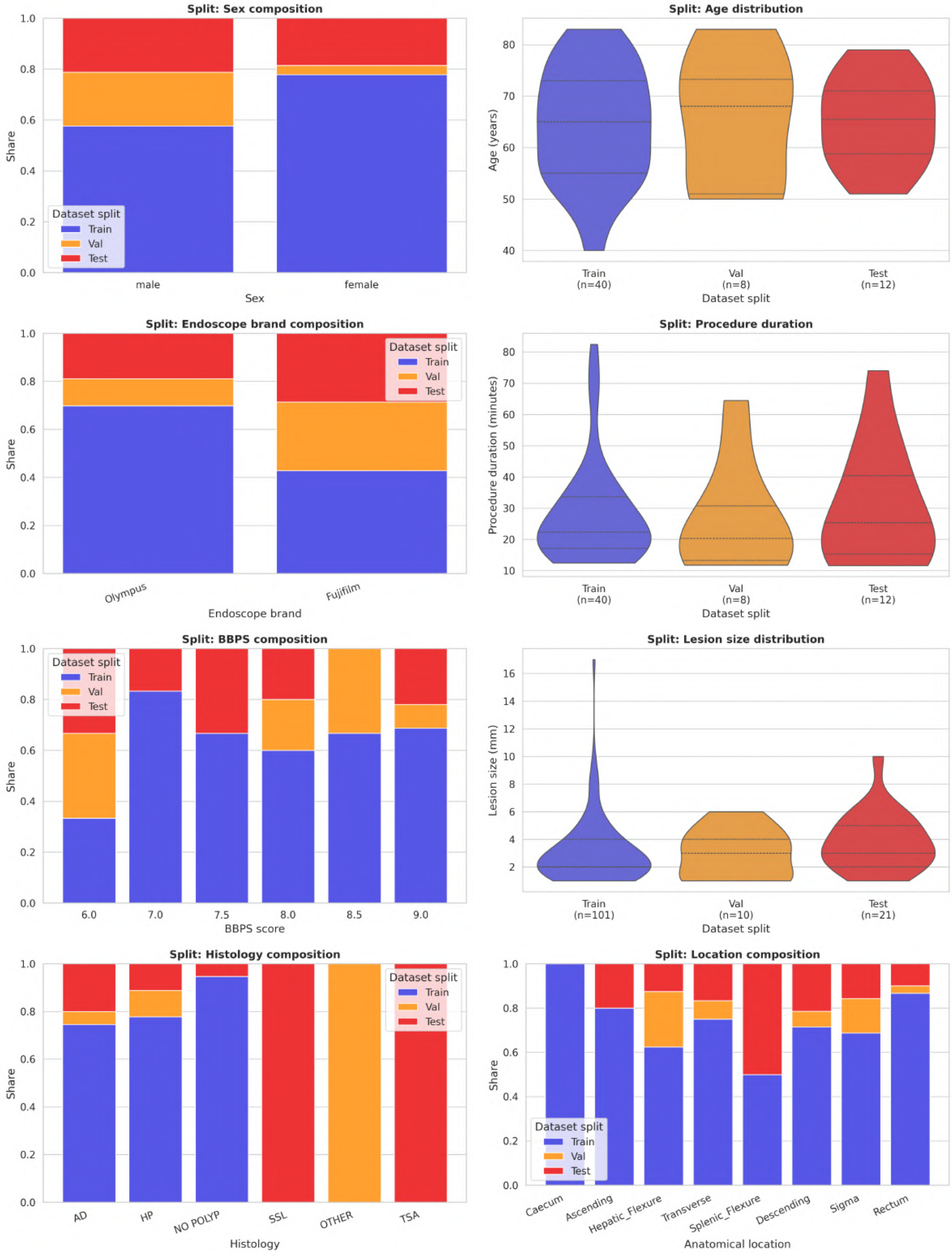


Figure 21: Lesion-level and preparation characteristics in the REAL-Colon dataset per split.

Table 19: SUN dataset: Case-level image counts and assigned data split. Positive cases have IDs 0-100 and negative cases 101-113.

Cases 1–57				Cases 58–113			
Case ID	Type	Split	#Images	Case ID	Type	Split	#Images
1	POS	train	527	58	POS	train	267
2	POS	train	1313	59	POS	train	646
3	POS	train	292	60	POS	train	146
4	POS	train	80	61	POS	train	679
5	POS	train	930	62	POS	train	351
6	POS	train	491	63	POS	train	632
7	POS	train	315	64	POS	train	81
8	POS	train	256	65	POS	train	222
9	POS	train	136	66	POS	train	1685
10	POS	train	436	67	POS	train	191
11	POS	train	113	68	POS	train	1319
12	POS	train	538	69	POS	train	130
13	POS	train	479	70	POS	train	264
14	POS	train	1183	71	POS	val	1021
15	POS	train	487	72	POS	val	774
16	POS	train	199	73	POS	val	1285
17	POS	train	304	74	POS	val	276
18	POS	train	243	75	POS	val	343
19	POS	train	96	76	POS	val	343
20	POS	train	3159	77	POS	val	215
21	POS	train	100	78	POS	val	267
22	POS	train	314	79	POS	val	76
23	POS	train	182	80	POS	val	1192
24	POS	train	973	81	POS	test	427
25	POS	train	338	82	POS	test	111
26	POS	train	370	83	POS	test	795
27	POS	train	249	84	POS	test	218
28	POS	train	195	85	POS	test	1393
29	POS	train	377	86	POS	test	257
30	POS	train	224	87	POS	test	454
31	POS	train	183	88	POS	test	249
32	POS	train	981	89	POS	test	149
33	POS	train	594	90	POS	test	479
34	POS	train	245	91	POS	test	1061
35	POS	train	1212	92	POS	test	391
36	POS	train	815	93	POS	test	452
37	POS	train	448	94	POS	test	136
38	POS	train	509	95	POS	test	606
39	POS	train	713	96	POS	test	301
40	POS	train	159	97	POS	test	431
41	POS	train	108	98	POS	test	170
42	POS	train	268	99	POS	test	161
43	POS	train	260	100	POS	test	188
44	POS	train	745	101	NEG	train	9960
45	POS	train	383	102	NEG	test	10073
46	POS	train	170	103	NEG	train	7152
47	POS	train	705	104	NEG	train	14635
48	POS	train	176	105	NEG	train	7916
49	POS	train	181	106	NEG	val	17046
50	POS	train	740	107	NEG	test	5636
51	POS	train	1737	108	NEG	train	2568
52	POS	train	207	109	NEG	train	9522
53	POS	train	245	110	NEG	train	7086
54	POS	train	345	111	NEG	test	4832
55	POS	train	700	112	NEG	val	6799
56	POS	train	248	113	NEG	test	6328
57	POS	train	326				
Total positive images			49,136	Total negative images			109,553
Total images			158,689				

A.3 Discussion

A.3.1 Baseline Experiment

Additional detection metrics and frame metrics calculated at a confidence per seed. For each seed (0, 42, 123), confidences for Faster R-CNN (0.10, 0.20, 0.26), and RT-DETR (0.32, 0.20, 0.32) are selected. The YOLO model confidences remain the earlier specified ones (0.06 for YOLOv8 and 0.05 for YOLOv11) across all seeds.

Table 20: Detection-level performance using multiple confidence thresholds.

Metric	**Faster R-CNN**	**YOLOv8**	**YOLOv11**	**RT-DETR**
Confidence 0.20				
precision	0.524 ± 0.086	0.701 ± 0.022	0.715 ± 0.011	0.398 ± 0.095
recall	0.463 ± 0.066	0.508 ± 0.022	0.453 ± 0.004	0.720 ± 0.041
Confidence Faster R-CNN: 0.20, YOLOv8: 0.06 YOLOv11: 0.05, RT-DETR: 0.30				
precision	0.524 ± 0.086	0.523 ± 0.029	0.516 ± 0.017	0.577 ± 0.094
recall	0.463 ± 0.066	0.605 ± 0.030	0.573 ± 0.009	0.651 ± 0.049
Confidence per seed				
precision	0.505 ± 0.012	0.523 ± 0.029	0.516 ± 0.017	0.537 ± 0.027
recall	0.470 ± 0.055	0.605 ± 0.030	0.573 ± 0.009	0.669 ± 0.006

Table 21: Frame-level performance using confidence thresholds per seed.

Metric	**Faster R-CNN**	**YOLOv8**	**YOLOv11**	**RT-DETR**
Confidence Faster R-CNN: 0.20, YOLOv8: 0.06 YOLOv11: 0.05, RT-DETR: 0.30				
sensitivity / TPR	0.463 ± 0.066	0.605 ± 0.030	0.573 ± 0.009	0.651 ± 0.049
specificity	0.954 ± 0.017	0.956 ± 0.006	0.957 ± 0.003	0.957 ± 0.017
FPR	0.046 ± 0.017	0.044 ± 0.006	0.043 ± 0.003	0.043 ± 0.017
Precision	0.637 ± 0.086	0.698 ± 0.023	0.693 ± 0.020	0.723 ± 0.070
F_1 score	0.532 ± 0.047	0.648 ± 0.015	0.627 ± 0.013	0.681 ± 0.001
F_2 score	0.488 ± 0.057	0.621 ± 0.024	0.594 ± 0.011	0.662 ± 0.031
Confidence per seed				
sensitivity	0.470 ± 0.055	0.605 ± 0.030	0.573 ± 0.009	0.669 ± 0.006
specificity	0.953 ± 0.003	0.956 ± 0.006	0.957 ± 0.003	0.952 ± 0.001
fpr	0.047 ± 0.003	0.044 ± 0.006	0.043 ± 0.003	0.048 ± 0.001
precision	0.625 ± 0.039	0.698 ± 0.023	0.693 ± 0.020	0.702 ± 0.005
F_1 score	0.536 ± 0.050	0.648 ± 0.015	0.627 ± 0.013	0.685 ± 0.003
F_2 score	0.494 ± 0.054	0.621 ± 0.024	0.594 ± 0.011	0.675 ± 0.005

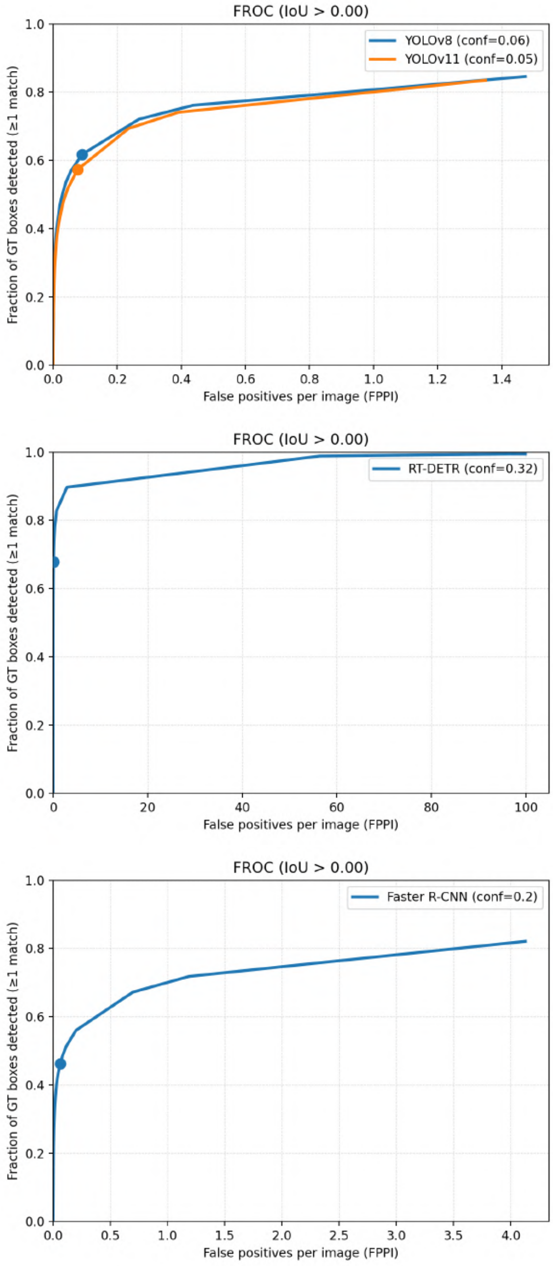
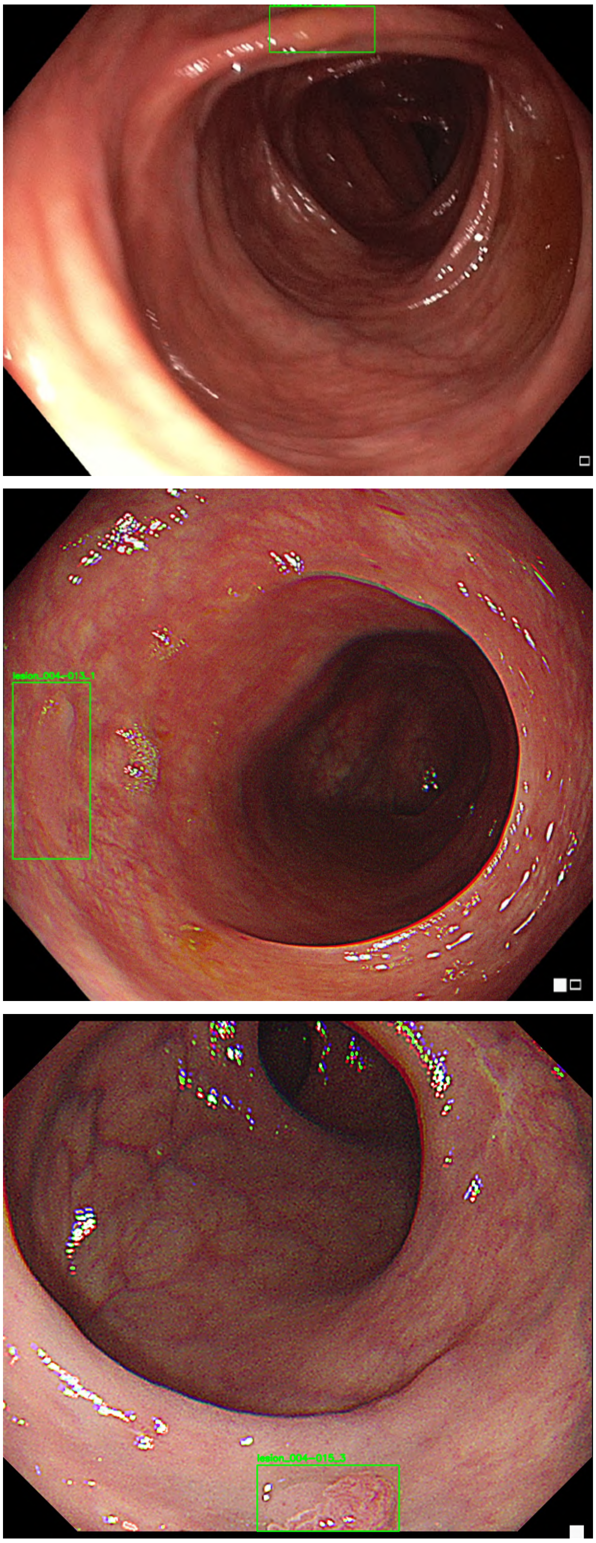


Figure 22: FROC curves for the evaluated detectors. From left to right: YOLO-based models (YOLOv8 and YOLOv11), RT-DETR, and Faster R-CNN.

The FROC curves are shown separately because the achievable false positives per image (FPPI) range differs substantially between models. Some detectors reach FPPI values that other models never attain, making a single combined plot misleading due to axis compression and loss of detail in the relevant operating regions. The 100 detections per frame match the restricted maximal detections for the RT-DETR output. The architecture always returns this fixed set, and typically only a small subset of predictions contains high confidence scores meaningful for analysis. For the other models the values are lower as predictions are pre-filtered using NMS.

A.3.2 Lesion Analysis



Lesion 7 (SSL)

Lesion 14 (SSL)

Lesion 21 (TSA)

Figure 23: Visual examples of histological subtypes in the test set (including both sessile serrated lesions (SSL), and a traditional serrated adenoma (TSA))

Bibliography

Carina Albuquerque, Roberto Henriques, and Mauro Castelli. Deep learning-based object detection algorithms in medical imaging: Systematic review. *Heliyon*, 11(1):e41137, 2025. ISSN 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2024.e41137. URL `https://www.sciencedirect.com/science/article/pii/S240584402417168X`.

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. Texture-based polyp detection in colonoscopy. In Hans-Peter Meinzer, Thomas Martin Deserno, Heinz Handels, and Thomas Tolxdorff, editors, *Bildverarbeitung für die Medizin 2009*, pages 346–350, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-540-93860-6.

J. Bernal, J. Sánchez, and F. Vilariño. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognition*, 45(9):3166–3182, 2012. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2012.03.002. URL `https://www.sciencedirect.com/science/article/pii/S0031320312001185`. Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011).

Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics*, 43:99–111, 2015. ISSN 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2015.02.007. URL `https://www.sciencedirect.com/science/article/pii/S0895611115000567`.

Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sánchez, Bogdan J. Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, Konstantin Pogorelov, Sungbin Choi, Quentin Debard, Lena Maier-Hein, Stefanie Speidel, Danail Stoyanov, Patrick Brandao, Henry Córdova, Cristina Sánchez-Montes, Suryakanth R. Gurudu, Gloria Fernández-Esparrach, Xavier Dray, Jianming Liang, and Aymeric Histace. Comparative validation of polyp detection methods in video colonoscopy: Results from the miccai 2015 endoscopic vision challenge. *IEEE Transactions on Medical Imaging*, 36(6):1231–1249, 2017. doi: 10.1109/TMI.2017.2664042.

Carlo Biffi, Giulio Antonelli, Sebastian Bernhofer, Cesare Hassan, Daizen Hirata, Mineo Iwatate, Andreas Maieron, Pietro Salvagnini, and Andrea Cherubini. Real-colon: A dataset for developing real-world ai applications in colonoscopy. *Scientific Data*, 11(1):539, 2024.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. URL `https://arxiv.org/abs/2004.10934`.

Michael Bretthauer, Magnus Løberg, Paulina Wieszczy, Mette Kalager, Louise Emilsson, Kjetil Garborg, Maciej Rupinski, Evelien Dekker, Manon Spaander, Marek Bugajski, Øyvind Holme, Ann G. Zauber, Nastazja D. Pilonis, Andrzej Mroz, Ernst J. Kuipers, Joy Shi, Miguel A. Hernán, Hans-Olov Adami, Jaroslaw Regula, Geir Hoff, and Michal F. Kaminski. Effect of colonoscopy screening on risks of colorectal cancer and related death. *New England Journal of Medicine*, 387(17):1547–1556, 2022. doi: 10.1056/NEJMoa2208375. URL `https://www.nejm.org/doi/full/10.1056/NEJMoa2208375`.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

Dev P. Chakraborty. A brief history of free-response receiver operating characteristic paradigm data analysis. *Academic Radiology*, 20(7):915–919, 2013. ISSN 1076-6332. doi: 10.1016/j.acra.2013.03.001. URL `https://www.sciencedirect.com/science/article/pii/S1076633213001104`.

Dev P. Chakraborty and Linda H. L. Winter. Free-response methodology: alternate analysis and a new observer-performance experiment. *Radiology*, 174(3):873–881, mar 1990. doi: 10.1148/radiology.174.3.2305073.

D. A. Corley, C. D. Jensen, A. R. Marks, W. K. Zhao, J. K. Lee, C. A. Doubeni, A. G. Zauber, J. de Boer, B. H. Fireman, J. E. Schottinger, V. P. Quinn, N. R. Ghai, T. R. Levin, and C. P. Quesenberry. Adenoma detection rate and risk of colorectal cancer and death. *New England Journal of Medicine*, 370(14):1298–1306, 2014. doi: 10.1056/NEJMoa1309086.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

Uxue Delaquintana-Aramendi, Leire Benito-del Valle, Aitor Alvarez-Gila, Javier Pascau, Luisa F Sánchez-Peralta, Artzai Picón, J Blas Pagador, and Cristina L Saratxaga. Ai-assisted colonoscopy: Polyp detection and segmentation using foundation models. *arXiv preprint arXiv:2503.24138*, 2025. URL `https://arxiv.org/abs/2503.24138`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0275-4. URL `https://doi.org/10.1007/s11263-009-0275-4`.

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. doi: 10.1109/TPAMI.2009.167.

Daniel Fitting, Adrian Krenzer, Joel Troya, Michael Banck, Boban Sudarevic, Markus Brand, Wolfgang Böck, Wolfram G. Zoller, Thomas Rösch, Frank Puppe, Alexander Meining, and Alexander Hann. A video based benchmark data set (endotest) to evaluate computer-aided polyp detection systems. *Scandinavian Journal of Gastroenterology*, 57(11):1397–1403, 2022. doi: 10.1080/00365521.2022.2085059. URL `https://doi.org/10.1080/00365521.2022.2085059`. PMID: 35701020.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. URL `https://arxiv.org/abs/2107.08430`.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7036–7045, 2019.

Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. doi: 10.1109/CVPR.2014.81.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, page 346–361. Springer International Publishing, 2014. ISBN 9783319105789. doi: 10.1007/978-3-319-10578-9_23.

Xin He and Eric C. Frey. ROC, LROC, FROC, AFROC: An Alphabet Soup. *Journal of the American College of Radiology*, 6(9):652–655, sep 2009. ISSN 1546-1440. doi: 10.1016/j.jacr.2009.06.001.

Priyanto Hidayatullah, Nurjannah Syakrani, Muhammad Rizqi Sholahuddin, Trisna Gelar, and Refdinal Tubagus. Yolov8 to yolo11: A comprehensive architecture in-depth comparative review, 2025. URL `https://arxiv.org/abs/2501.13400`.

Sae Hwang, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C. de Groen. Polyp detection in colonoscopy video using elliptical shape feature. In *2007 IEEE International Conference on Image Processing*, volume 2, pages II – 465–II – 468, 2007. doi: 10.1109/ICIP.2007.4379193.

Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D. Johansen. Kvasir-seg: A segmented polyp dataset, 2019. URL `https://arxiv.org/abs/1911.07069`.

Glenn Jocher. Ultralytics yolov5, 2020. URL `https://github.com/ultralytics/yolov5`. Version 5.0.0, AGPL-3.0 license, accessed 15 Nov 2025.

Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL `https://github.com/ultralytics/ultralytics`. Version 11.0.0, AGPL-3.0 license, accessed 15 Nov 2025.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL `https://github.com/ultralytics/ultralytics`. Version 8.0.0, AGPL-3.0 license, accessed 15 Nov 2025.

J. Kang and R. Doraiswami. Real-time image processing system for endoscopic applications. In *CCECE 2003 - Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436)*, volume 3, pages 1469–1472 vol.3, 2003. doi: 10.1109/CCECE.2003.1226181.

Uri Ladabaum, Jason A. Dominitz, Charles Kahi, and Robert E. Schoen. Strategies for colorectal cancer screening. *Gastroenterology*, 158(2):418–432, 01 2020. doi: 10.1053/j.gastro.2019.06.043.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178, 2006. doi: 10.1109/CVPR.2006.68.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Ji Young Lee, Jinhoon Jeong, Eun Mi Song, Chang Ha, Hye-Jeong Lee, Ja-Eun Koo, Dong-Hoon Yang, Nam Kim, and Jeong-Sik Byeon. Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets. *Scientific Reports*, 10(1):8379, May 2020. doi: 10.1038/s41598-020-65387-1. URL `https://doi.org/10.1038/s41598-020-65387-1`. PMID: 32433506; PMCID: PMC7239848.

Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3.0: A full-scale reloading, 2023. URL `https://arxiv.org/abs/2301.05586`.

Kaidong Li, Mohammad I Fathan, Krushi Patel, Tianxiao Zhang, Cuncong Zhong, Ajay Bansal, Amit Rastogi, Jean S Wang, and Guanghui Wang. Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations. *Plos one*, 16(8):e0255809, 2021.

Zijun Liao, Yian Zhao, Xin Shan, Yu Yan, Chang Liu, Lei Lu, Xiangyang Ji, and Jie Chen. Rt-detrv4: Painlessly furthering real-time object detection with vision foundation models, 2025. URL `https://arxiv.org/abs/2510.25257`.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1. URL `https://cocodataset.org/`.

Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. URL `https://api.semanticscholar.org/CorpusID:10716717`.

Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128, 02 2020. doi: 10.1007/s11263-019-01247-4.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. URL `https://arxiv.org/abs/2303.05499`.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation, 2018. URL `https://arxiv.org/abs/1803.01534`.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016. ISBN 9783319464480. doi: 10.1007/978-3-319-46448-0_2. URL `http://dx.doi.org/10.1007/978-3-319-46448-0_2`.

Yiming Liu, Ping Xu, Tian Wang, Ming Wang, Li Cao, and Mingmin Li. Data augmentation and improved deformable detr for polyp detection. In *2023 China Automation Congress (CAC)*, pages 9114–9118, 2023. doi: 10.1109/CAC59555.2023.10451257.

Yiting Ma, Xuejin Chen, Kai Cheng, Yang Li, and Bin Sun. Ldpolypvideo benchmark: A large-scale colonoscopy video dataset of diverse polyps. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27 – October 1, 2021, Proceedings, Part V*, page 387–396, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-87239-7. doi: 10.1007/978-3-030-87240-3_37. URL `https://doi.org/10.1007/978-3-030-87240-3_37`.

N. Mahmud, J. Cohen, K. Tsourides, and T. M. Berzin. Computer vision and augmented reality in gastrointestinal endoscopy. *Gastroenterology Report*, 3(3): 179–184, 2015. doi: 10.1093/gastro/gov027.

Masashi Misawa, Shin ei Kudo, Yuichi Mori, Kinichi Hotta, Kazuo Ohtsuka, Takahisa Matsuda, Shoichi Saito, Toyoki Kudo, Toshiyuki Baba, Fumio Ishida, Hayato Itoh, Masahiro Oda, and Kensaku Mori. Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video). *Gastrointestinal Endoscopy*, 93(4):960–967.e3, 2021. ISSN 0016-5107. doi: https://doi.org/10.1016/j.gie.2020.07.060. URL `https://www.sciencedirect.com/science/article/pii/S0016510720346551`.

Eileen Morgan, Melina Arnold, A Gini, V Lorenzoni, C J Cabasag, Mathieu Laversanne, Jerome Vignat, Jacques Ferlay, Neil Murphy, and Freddie Bray. Global burden of colorectal cancer in 2020 and 2040: incidence and mortality estimates from globocan. *Gut*, 72(2):338–344, 2023. ISSN 0017-5749. doi: 10.1136/gutjnl-2022-327736. URL `https://gut.bmj.com/content/72/2/338`.

Bernd Münzer, Klaus Schoeffmann, and Laszlo Böszörmenyi. Content-based processing and analysis of endoscopic images and videos: A survey. *Multimedia Tools and Applications*, 77(1):1323–1362, jan 2018. doi: 10.1007/s11042-016-4219-z. URL `https://doi.org/10.1007/s11042-016-4219-z`.

A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855, 2006. doi: 10.1109/ICPR.2006.479.

M. Y. Nie, X. W. An, Y. C. Xing, Z. Wang, Y. Q. Wang, and J. Q. Lü. Artificial intelligence algorithms for real-time detection of colorectal polyps during colonoscopy: A review. *American Journal of Cancer Research*, 14(11):5456–5470, 2024. doi: 10.62347/BZIZ6358.

I. Pacal, A. Karaman, D. Karaboga, B. Akay, A. Basturk, U. Nalbantoglu, and S. Coskun. An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets. *Computers in Biology and Medicine*, 141:105031, 2022. doi: 10.1016/j.compbiomed.2021.105031. URL `https://doi.org/10.1016/j.compbiomed.2021.105031`.

Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020. doi: 10.1109/IWSSIP48289.2020.9145130.

Participants in the Paris Workshop. The paris endoscopic classification of superficial neoplastic lesions: esophagus, stomach, and colon: November 30 to december 1, 2002. *Gastrointestinal Endoscopy*, 58(6, Supplement):S3–S43, 2003. ISSN 0016-5107. doi: https://doi.org/10.1016/S0016-5107(03)02159-X. URL `https://www.sciencedirect.com/science/article/pii/S001651070302159X`.

Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2016. URL https://api.semanticscholar.org/CorpusID:786357.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. URL https://arxiv.org/abs/1804.02767.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015. URL http://arxiv.org/abs/1506.02640.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

Douglas K. Rex, Cesare Hassan, and Michael J. Bourke. The colonoscopist's guide to the vocabulary of colorectal neoplasia: histology, morphology, and management. *Gastrointestinal Endoscopy*, 86(2):253–263, 2017. ISSN 0016-5107. doi: https://doi.org/10.1016/j.gie.2017.03.1546. URL https://www.sciencedirect.com/science/article/pii/S0016510717317844.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. pages 658–666, 06 2019. doi: 10.1109/CVPR.2019.00075.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Neural architecture search for real-time detection transformers, 2025. URL https://arxiv.org/abs/2511.09554.

Ranjan Sapkota, Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, Hong Yan, and Manoj Karkee. Yolo advances to its genesis: a decadal and comprehensive review of the you only look once (yolo) series. *Artificial Intelligence Review*, 58, 06 2025. doi: 10.1007/s10462-025-11253-3.

Juan Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International Journal of Computer Assisted Radiology and Surgery*, 9 (2):283–293, mar 2014. ISSN 1861-6429. doi: 10.1007/s11548-013-0926-3. URL https://doi.org/10.1007/s11548-013-0926-3.

Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. ISSN 0306-4573. doi: 10.1016/j.ipm.2009.03.002. URL https://www.sciencedirect.com/science/article/pii/S0306457309000259.

Yibo Sun, Zhe Sun, and Weitong Chen. The evolution of object detection methods. *Engineering Applications of Artificial Intelligence*, 133:108458, 2024. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2024.108458. URL `https://www.sciencedirect.com/science/article/pii/S095219762400616X`.

Luisa F. Sánchez-Peralta, J. Blas Pagador, Artzai Picón, Ángel José Calderón, Francisco Polo, Nagore Andraka, Roberto Bilbao, Ben Glover, Cristina L. Saratxaga, and Francisco M. Sánchez-Margallo. Piccolo white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets. *Applied Sciences*, 10(23), 2020. ISSN 2076-3417. doi: 10.3390/app10238501. URL `https://www.mdpi.com/2076-3417/10/23/8501`.

Nima Tajbakhsh, Suryakanth R. Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging*, 35(2):630–644, 2016. doi: 10.1109/TMI.2015.2487997.

Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4): 1680–1716, November 2023. ISSN 2504-4990. doi: 10.3390/make5040083. URL `http://dx.doi.org/10.3390/make5040083`.

J. Troya, B. Sudarevic, A. Krenzer, M. Banck, M. Brand, B. M. Walter, F. Puppe, W. G. Zoller, A. Meining, and A. Hann. Direct comparison of multiple computer-aided polyp detection systems. *Endoscopy*, 56(1):63–69, Jan 2024. doi: 10.1055/a-2147-0571. PMID: 37532115; PMCID: PMC10736101.

Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. doi: 10.1007/s11263-013-0620-5.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Comput. Vis. Pattern Recog*, 1, 01 2001.

Jing-Jing Wan, Peng-Cheng Zhu, Bo-Lun Chen, and Yong-Tao Yu. A semantic feature enhanced yolov5-based network for polyp detection from colonoscopy images.

Scientific Reports, 14(1):15478, Jul 2024a. doi: 10.1038/s41598-024-66642-5. URL `https://doi.org/10.1038/s41598-024-66642-5`.

Jingjing Wan, Wenjie Zhu, Bolun Chen, Ling Wang, Kailu Chang, and Xianchun Meng. Crh-yolo for precise and efficient detection of gastrointestinal polyps. *Scientific Reports*, 14(1):30033, Dec 2024b. doi: 10.1038/s41598-024-81842-9. URL `https://doi.org/10.1038/s41598-024-81842-9`. PMID: 39626388.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. URL `https://arxiv.org/abs/2405.14458`.

Chien-Yao Wang and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024. URL `https://arxiv.org/abs/2402.13616`.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022. URL `https://arxiv.org/abs/2207.02696`.

Pu Wang, Tyler M Berzin, Jeremy R Glissen Brown, Shai Bharadwaj, Aymeric Becq, Xiao Xiao, Peixi Liu, Lishun Li, Yi Song, Di Zhang, Yiyang Li, Guojian Xu, Meng Tu, and Xiaogang Liu. Real-time automatic detection system increases colonoscopic polyp and adenoma detection rates: a prospective randomised controlled study. *Gut*, 68(10):1813–1819, oct 2019. doi: 10.1136/gutjnl-2018-317500.

Sidney J. Winawer, Ann G. Zauber, May Nah Ho, Michael J. O'Brien, Leonard S. Gottlieb, Stephen S. Sternberg, Jerome D. Waye, Melvin Schapiro, John H. Bond, Joel F. Panish, Frederick Ackroyd, Moshe Shike, Robert C. Kurtz, Lynn Hornsby-Lewis, Hans Gerdes, and Edward T. Stewart. Prevention of colorectal cancer by colonoscopic polypectomy. *New England Journal of Medicine*, 329(27):1977–1981, 1993. doi: 10.1056/NEJM199312303292701. URL `https://www.nejm.org/doi/full/10.1056/NEJM199312303292701`.

Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior, 2021. URL `https://arxiv.org/abs/2104.01318`.

Youngbeom Yoo, Jae Young Lee, Dong-Jae Lee, Jiwoon Jeon, and Junmo Kim. Real-time polyp detection in colonoscopy using lightweight transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7809–7819, January 2024.

Xu Zhang, Fei Chen, Tao Yu, Jiye An, Zhengxing Huang, Jiquan Liu, Weiling Hu, Liangjing Wang, Huilong Duan, and Jianmin Si. Real-time gastric polyp detection using convolutional neural networks. *PLOS ONE*, 14(3):1–16, 03 2019. doi: 10.1371/journal.pone.0214133. URL `https://doi.org/10.1371/journal.pone.0214133`.

S. Zhao, S. Wang, P. Pan, T. Xia, X. Chang, X. Yang, L. Guo, Q. Meng, F. Yang, W. Qian, Z. Xu, Y. Wang, Z. Wang, L. Gu, R. Wang, F. Jia, J. Yao, Z. Li, and Y. Bai. Magnitude, risk factors, and factors associated with adenoma miss rate of tandem colonoscopy: A systematic review and meta-analysis. *Gastroenterology*, 156(6):1661–1674.e11, 2019a. doi: 10.1053/j.gastro.2019.01.260.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16965–16974, 2024. URL `https://github.com/lyuwenyu/RT-DETR`. Ultralytics implementation: `https://docs.ultralytics.com/models/rtdetr/`.

Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–21, 01 2019b. doi: 10.1109/TNNLS.2018.2876865.

Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. *arXiv preprint arXiv:1911.08287*, 2019. URL `https://arxiv.org/abs/1911.08287`.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. URL `https://arxiv.org/abs/2010.04159`.

Declaration of Authorship

Ich erkläre hiermit gemäß §9 Abs. 12 APO, dass ich die vorstehende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Abschlussarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Bamberg , 2.1.2026

Place, Date

Andreas Schuus

Signature