Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Real-Time Polyp Detection System with Clinical Application in Colonoscopy Using Deep Convolutional Neural Networks

Adrian Krenzer (adrian.krenzer@uni-wuerzburg.de)
University of Würzburg
Michael Banck
University of Würzburg
Kevin Makowski
University of Würzburg
Amar Hekalo
University of Würzburg
Daniel Fitting
University Hospital Würzburg
Joel Troya
University Hospital Würzburg
Boban Sudarevic
Katharinenhospital
Wolfram G. Zoller
Katharinenhospital
Alexander Hann
University Hospital Würzburg
Frank Puppe
University of Würzburg

Research Article

Keywords: Machine learning, Deep learning, Endoscopy, Gastroenterology, Automation, Object detection, Video object detection, Real-time

Posted Date: February 8th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1310139/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

RESEARCH

A real-time polyp detection system with clinical application in colonoscopy using deep convolutional neural networks

Adrian Krenzer[1,2*], Michael Banck[1,2], Kevin Makowski[1], Amar Hekalo[1], Daniel Fitting[2], Joel Troya[2], Boban Sudarevic[2,3], Wolfram G. Zoller[3], Alexander Hann[2] and Frank Puppe[1]

[*]Correspondence: adrian.krenzer@uni-wuerzburg.de
[1]Department of Artificial Intelligence and Knowledge Systems, Sanderring 2, 97070 Würzburg, Germany
Full list of author information is available at the end of the article

Abstract

Background: Colorectal cancer (CRC) is still a leading cause of cancer-related deaths worldwide. The best method to prevent CRC is a colonoscopy. During this procedure, the colonoscopist searches for polyps. However, there is a potential risk of polyps being missed by the examiner. Here the automated detection of polyps helps assist the examiner during coloscopy. In the literature, there are already publications examining the problem of polyp detection. Nevertheless, most of these systems are only used in the research context and do not attain clinical application. Therefore, we introduce a system scoring best on current benchmarks and implementing it fully for clinical-ready applications.

Methods: To create the polyp detection system (ENDOMIND-Advanced), we combined our own collected data from different hospitals and practices in Germany with open-source data sets to create a data set with over 500.000 annotated images. Furthermore, we show different techniques for training a CNN on polyp detection that involves preprocessing, data augmentation and hyperparameter optimization. Additionally, we developed a post-processing technique based on video detection to work in real-time with a stream of images. This allows us to leverage the incoming stream context of the endoscope while maintaining real-time performance. Furthermore, the polyp detection system is integrated into a prototype ready for application in clinical interventions.

Results: First, we show that our polyp detection system is state of the art by evaluating it on the CVC-VideoClinicDB benchmark with a F1-score of 90.24%. We compare the polyp detection system to the best system in the literature and achive better results in speed and accuracy. Additionally, we show its performance on our own data and introduce a new metric called the time to the first detection. This metric is given in seconds and shows how long AI systems need to detect a polyp for the first time. Finally, we further elaborate on the explainability of our system by showing heatmaps of the neural network explaining neural activations.

Conclusion: Overall we introduce a fully assembled real-time system for polyp detection with application in clinical practice and show that the system outperforms current systems on benchmark data sets with real-time performance.

Keywords: Machine learning; Deep learning; Endoscopy; Gastroenterology; Automation; Object detection; Video object detection; Real-time

Background

Colorectal cancer (CRC) is the second leading cause of cancer-related deaths worldwide [1]. One of the best methods to avoid CRC is to perform a colonoscopy to detect

the potential disease as early as possible. In a colonoscopy the large intestine (colon) is examined with a long flexible tube that is inserted into the rectum. A small camera is mounted at the end of the tube, enabling the physician to look inside the colon [2]. During this procedure the colonoscopist searches for polyps and examines them closely. Polyps are protrusions of the mucosal surface of various shapes and sizes that can be benign or malignant and thus can develop into colorectal cancer. Polyps grow on the lining of the colon, which often does not cause symptoms. The two main types are non-neoplastic and neoplastic polyps. Non-neoplastic polyps usually do not become cancerous and polyps of type neoplastic might become cancerous [3]. This means that even if many polyps are not cancerous, some can turn into colon cancer. Ideally, the colonoscopist detects every polyp during a colonoscopy and decides on closer inspection whether it needs to be removed. Nevertheless, there is still a potential risk of polyps being missed. It has been shown that up to 27% of diminutive polyps are overlooked by physicians [4] [5], which happens due to lack of experience or fatigue. It has also been shown that even a general error rate of 20%-24% leads to a high risk for patients to die from CRC [6] [7]. Two studies have shown that the missing rate is related to the size of the polyp. Kim et al. showed that polyps of size $\leq$5 mm, 5-10 mm and $\geq$10 mm had a missing rate of 35.4%, 18.8% and 4.9%, respectively [8] . Ahn et al. demonstrated missing rates of 22.9%, 7.2% and 5.8% for sizes of $\leq$5 mm, 5-10 mm and $\geq$10 mm, respectively [9]. Both studies also found that the missing rate was higher when the patient had more than one polyp. Additionally, a systematic review calculated a similar value and received missing rates of 2%, 13%, and 26% for polyp sizes of $\geq$10 mm, 5-10 mm and 1-5 mm, respectively [10]. This indicates that smaller polyps have a higher risk of being missed by the colonoscopist. Missed polyps can have fatal consequences for the patient. Thus, the colonoscopist must detect and afterwards remove all potential cancerous polyps in order to minimize the risk of colorectal cancer [9].

To avoid missing polyps computer science research methods are developed to assist physicians during colonoscopy. The use of computers to detect polyps is called *computer-aided detection (CAD)*. The research field already has publications examining the problem of polyp detection. Nevertheless, most of these systems are only used in research context and do not attain actual clinical application. Therefore, we introduce a system scoring best on current benchmarks and implementing it fully for clinical-ready applications. The main contributions of our paper are:

1) *We introduce a fully assembled real-time system for polyp detection with application in clinical practice.*
2) *We show that the system outperforms current systems on benchmark data sets with real-time performance.*
3) *We introduce a novel post processing method working in real-time based on REPP [11] and use a new metric for polyp detection, which has value for clinical usage.*

Additionally, the polyp detection system was publicly funded and developed by computer engineers as well as endoscopists in the same workgroup to ensure high quality of the polyp detections. Figure 1 shows results of the polyp detection system.

To overview existing work and properly allocate our paper in the literature, we describe a brief history reaching from general polyp detection with handcrafted features up to state-of-the-art polyp detection with deep learning techniques.

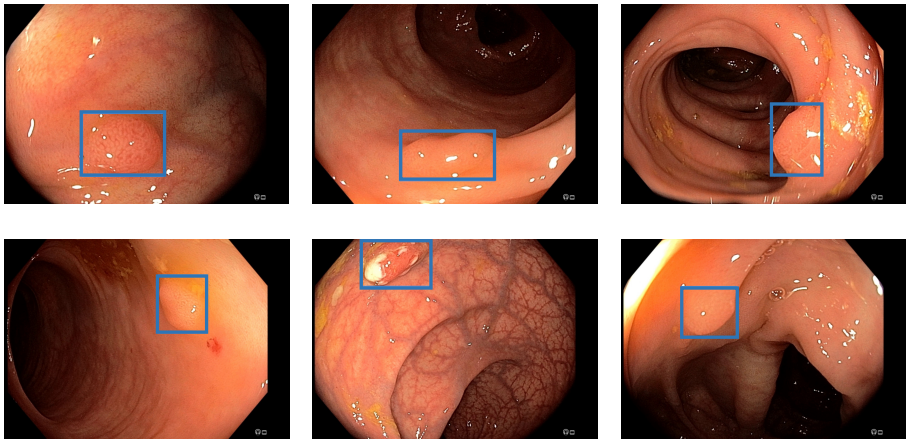


Figure 1 Detection examples. This figure illustrated some detection examples of the polyp detection system on our own data (EndoData).

A brief history of computer-aided polyp detection

One method to assist physicians is *computer-aided detection (CAD)*, the use of computers to detect polyps. The field of CAD is divided into two subfields CADe and CADx. CADe deals with the detection and localisation of polyps and CADx deals with the characterizations of polyps. This paper will focus exclusively on the CADe area. In this section, we only consider methods that localise polyps by specifying a rectangular section of the screen, a *bounding box*. Methods with a more precise pixel-wise localisation, which also detect the specific shape of a polyp, are not considered. Of course, given a segmentation, a bounding box around it can be easily computed, so the segmentation should be thought of as an extension of the localisation.

Computer-aided detection with handcrafted features The first approaches for computer-aided detection of polyps were explored as early as the late 1990s. For example, Krishnan et al. proposed the use of curvature analysis to detect polyps by shape [12]. Another method was presented in 2003 by Karkanis et al. They used wavelet transform to detect polyps based on their color and texture [13]. Hwang et al. used a new technique to distinguish eliptical shape features of polyp regions from non-polyp regions. They compared the features based on curvature, intensity, curve direction, and distance from the edge [14]. Bernal et al. (2012) proposed another method by converting images of polyps to grayscale so that the elevations of the polyps could be seen. Subsequently, the authors illuminated the outlines of the polyps, which they termed valleys. Based on the intensity of the valleys, the polyps were extracted and localised [15]. Furthermore, expert knowledge was used to hand-craft rules for detecting polyps based on certain properties such as size,

Table 1 Overview of polyp detection algorithms on still image data sets.

Author	Year	Method	Test data set	F1-score	Speed
Yuan et al.[19]	2020	DenseNet-UDCS	Custom	81.83%	N/A
Liu et al.[20]	2020	ADGAN	Custom	72.96%	N/A
Wang et al.[21]	2019	CenterNet	CVC-ClinicDB	97.88%	52 FPS
Liu et al.[22]	2019	SSD	CVC-ClinicDB	78.9%	30 FPS
Zhang et al.[23]	2019	SSD	ETIS-Larib	69.8	24 FPS
Zheng et al.[24]	2018	YOLO	ETIS-Larib	75.7%	16 FPS
Mo et al.[25]	2018	Faster R-CNN	CVC-ClinicDB	91.7%	17 FPS

shape, and color. Newer examples of these can be found in [16] and [17], both of which use SVMs. Additionally, real-time detection with handcrafted features was tested in clinical application [18]. The authors used a weighted combination of color, structure, textures, and motion information to detect image areas where a polyp is possibly located. The detection rate was 73%. Nevertheless, the rise of CNN-based methods in the field of image processing has superseded all of these techniques, as CNN methods have proven to show better results.

Methods involving CNNs Computer-aided polyp recognition was particularly shaped by various deep learning methods from the beginning of the last decade. We listed an overview of the essential algorithms on still image data sets in table 1. Specifically, a great deal of research interest has developed for the object recognition capabilities of Convolutional Neural Networks. For example, in 2015, authors Zhu et al. presented a seven-layer CNN as a feature extractor with a support vector machine (SVM) as a classifier to detect anomalies in endoscopy images [26]. The system was trained on custom data. The earlier approaches considered using an existing CNN architecture to localise polyps, the AlexNet [27][28][29]. This was developed for general image classification, i.e., not specifically for the medical field. The paper by Tajbakhsh et al. [28] states that the AlexNet [27] for polyp detection is better not trained completely, i.e., starting from random weights, but the already pre-trained weights should be used. It is shown that *transfer learning* is an effective approach in the presence of limited data, as generally given in the medical field.

Yuan et al. [29] first extract an interesting image section via edge-finding algorithms as a preliminary step and use it as input to the AlexNet [27]. This resulted in a high recall of 91.76% compared to the state of the art at that time. Mo et al. [25] are the first to use the unmodified *Faster R-CNN* [30] architecture for polyp detection. This allows the detection of polyps that are mostly obscured or very close to the camera, in particular, unlike previous models. The model is trained on the CVC-ClinicDB data. The model is robust to illumination changes or circular bubbles, but it misses some smaller polyps and sometimes gets too guided by an oval shape, increasing the number of FP. The authors also wanted to focus on these problems in the future. Shin et al. [31] were the first to use the *Inception-Resnet* [32] architecture unmodified for polyp detection. The model is trained on the ASU-Mayo-Video-DB.

They also added two post-processing methods, *false positive learning* and *offline learning*, to further improve the performance of the model. The advantage of the model was that the entire frame could be used for training, rather than a previous

patch extraction step. One problem with the model, as with Mo et al, is the high number of FP triggered by polyp-like structures. The authors plan to focus on improving the speed in the future, which was only 2.5 FPS. Zheng et al. [24] use the unmodified *YOLO* architecture [33]. Again, the advantages are that there is only one processing step, so no preliminary step to extract an RoI. As a result, the model was faster compared to two-step methods, but still only achieved 16 FPS. Further, the authors note that the CNN features of *white light* and *narrow-band* images differed greatly and thus they should be considered separately. The model is trained on the CVC-CLinicDB, CVC-ColonDB and custom data. Liu et al. [22] implemented and compared different backend models as feature extractors for the *single shot detection* architecture (SSD) [34]. These were *ResNet50* [35], *VGG16* [36], and *InceptionV3* [37], with InceptionV3 showing the best balanced result. Advantages of the models are robustness to size and shape, as well as speed, which is real-time capable at 30 FPS. The models is trained on the CVC-CLinicDB, CVC-ColonDB and ETIS-Larib data. In the future, other backend models could result in a further boost in performance.

Zhang et al. [38] used the *SSD-GPNet*, this is based on the SSD architecture [34], but tries to incorporate information that is normally lost by the standard pooling layers into the result through various customized pooling methods. Since it is based on the SSD architecture, this method is also fast and achieves real-time capability at 50 FPS; it also achieves good recall, especially for small polyps. For the future, the authors want to test their approaches for other diseases and find more ways to use as much of the image information as possible without increasing the complexity of the models. Furthermore, Zhang et al. presented another Deep Learning method for polyp detection and localisation. They presented a special single-shot multibox detector-based CNN model that reused displaced information through max-pooling layers to achieve higher accuracy. At 50 frames per second, the method provided real-time polyp detection while achieving a mean average precision of up to 90.4% [39]. The model is trained on custom data. Authors Bagheri et al. staged a different idea in which they first converted the input images into three color channels and then passed them to the neural network. This is to allow the network to learn correlated information using the preprocessed information about the color channels to locate and segment polyps [40]. With the same goal, Sornapudi et al. in their paper used region-based CNNs to localise polyps in colonoscopy images and in wireless capsule endoscopy (WCE) images. During localisation, images were segmented and detected based on polyp-like pixels [41].

In addition to CNNs, research is also being conducted on other deep learning methods for polyp detection. For example, a special sparse autoencoder method called stacked sparse autoencoder with image manifold constraint was used by Yuan and Meng [42] to detect polyps in WCE images. A sparse autoencoder is an artificial neural network commonly used for Unsupervised Learning methods [43]. The particular sparse autoencoder achieved 98% accuracy in polyp detection [42]. The system is trained and tested on the ASU-Mayo-Video-DB. Wang et al. [21] used the *AFP-Net*. Unlike an SSD model, an AFP-Net model does not require predefined anchor boxes, it is *anchor free*. It was the first application of such an architecture for polyp detection. Through *context enhancement module* (CEM), a *cosine ground-truth projection* and a customized loss function, the speed was increased and 52.6

FPS was achieved, which is real-time capable. For the future, the authors still want to improve the detection of the hard-to-detect small and flat polyps. The model is trained on the CVC-ClinicVideoDB. Liu et al. [20] used an *anomaly detection generative adversarial network* (ADGAN), which is based on the WGAN [44]. The goal of ADGAN is to learn only based on healthy images without polyps to reconstruct them. If this model receives an image with polyp as input, the model cannot reconstruct it, so at this point in the output there is a noticeably large difference from the input, which is easy to check. The problem of connecting the input to the latency space of the GAN was solved by a second GAN. In addition, a new loss function was added to improve performance even further. The model is trained on custom data.

The advantage of this approach is that no costly annotated data sets are needed and significantly larger amounts of data of normal intestinal mucosa are available. For the future, the authors want to make sure that frames with disturbances of a biological nature, such as stool residue or water, are also processed well, since these are often sorted out beforehand from many data sets. Yuan et al. [19] use the *DenseNet-UDCS* architecture for frame classification, not localisation, of *wireless capsule endoscopy* (WCE) images. The DenseNets [45] structure is kept unchanged, but the loss function is adapted. On the one hand, a weighting is introduced to compensate for the large imbalance in the size of the classes (without or with polyp). On the other hand, the loss function is adapted to be class sensitive, i.e., it forces that for the same class also similar features are learned and the features of the other class have as large differences as possible. These adaptations lead to improved performance and can be easily applied to other applications and models. For the future, the researchers still want to find a way to compensate for different illuminations by pre-processing methods and test attention-based methods. Another method is to use transformers in combination with CNNs. Zhang et al. used in parallel the ability of viewing global information of the whole image through the attention layers of the transformers as well as the detailed local detection of the CNNs to efficiently segment polyps. In addition, a new fusion technique called BiFusion was used to fuse the features obtained by the transformers and the CNNs. The resulting method called TransFuse stood out mainly because of its segmentation time of 98.7 FPS (frames per second) [46]. The model is trained on custom data.

3D and temporal methods While older publications are evaluated on still image benchmarks like the CVC-ClinicDB. The new state of the art is evaluated on the more challenging and more realistic video data set like CVC- VideoClinicDB. E.g. Wang et al. [21] have a high score of 97.88% on the CVC-ClinicDB data set. Nevertheless, this data set only involves 612 still images. We reconstructed the algorithm of Wang et al. [21] but could not reproduce the results on the video data sets. For these video data sets, all frames are extracted and polyps in these frames are annotated with corresponding bounding boxes. We listed an overview of the essential algorithms on video data sets in table 2. Another, still little explored, approach is to use the temporal information within the video. In the above mentioned methods only single frames are considered, thus information which is given by the sequence of the frames is lost. In Itoh et al. [50], temporal information is included through

Table 2 Overview of polyp detection algorithms on video data sets.

Author	Year	Method	Test data set	F1-score	Speed
Xu et al.[47]	2021	CNN + ISTM	CVC-VideoClinicDB	75.86%	N/A
Qadir et al.[48]	2020	Faster R-CNN	CVC-VideoClinicDB	84.44%	15 FPS
		SSD	CVC-VideoClinicDB	71.82%	33 FPS
Yuan et al.[19]	2020	DenseNet-UDCS	Custom	81.83%	N/A
Zhang et al.[38]	2019	SSD-GPNet	Custom	84.24%	50 FPS
Misawa et al.[49]	2019	3D-CNN	Custom	N/A	N/A
Itoh et al.[50]	2019	3D-ResNet	Custom	N/A	N/A
Shin et al.[31]	2018	Inception ResNet	ASU-Mayo-Video-DB	86.9%	2.5 FPS
Yuan et al.[29]	2017	AlexNet	ASU-Mayo-Video-DB	N/A	N/A
Tajbakhsh et al.[28]	2016	AlexNet	Custom	N/A	N/A

a *3D-ResNet*. In addition, a weighted loss function and selection of so-called *hard negative frames* address the problem of training data class imbalance. These lead to an improvement of 2% F1-score. However, one problem is that the model overfitted more easily than its 2D counterpart because it has more parameters and is not applicable in real-time. Zhang et al. [23] combine the output of a conventional SSD model [34] via a *Fusion module* with a generated *Optical Flow*. This is similar to a heat map showing motion over short periods of time and is easy to compute. This approach is much less complex and therefore faster compared to other temporal systems that use 3D methods still it is not applicable for real-time polyp detection. Misawa et al. [49] use a *3D-CNN* to include temporal information. This allows many different types of polyps to be well detected. The model is trained on custom data.

Additionally, Qadir et al. [48] use a conventional localisation model, such as SSD [34] or Faster R-CNN [30], and further process the output of these through an *FP Reduction Unit*. This looks at the position of the generated bounding boxes over the 7 preceding and following frames and tries to find and possibly correct outliers. Because future frames are used, there is a small delay, but the actual calculation of the *FP Reduction Unit* is fast. A different and promising method was provided by Qadir et al. in a two-step process. They used a CNN in the first step, which generated several regions of interest (RoIs) for classification. Then, these proposed RoIs were compared based on the subsequent frames and their RoIs and classified into True Positive and False Positive. The assumption of this method is that the frame in a video should be similar to the next frame. It intends to reduce the percentage of false predictions [51]. Because CNNs are sensitive to noise in the data they may produce a high count of false positives. Another approach is therefore using a two-stage method that first suggest multiple region of interests (RoI). Then, the current proposed RoIs are categorized as true positive and false positives by considering the RoIs of the following frames [48]. With this method they are reducing the number of false positives and reaching state-of-the-art results. The model is trained on the ASU-Mayo-Video-DB and custom data. Moreover, Xu et al. [47] designed a 2D CNN detector including spatiotemporal information involving a ISTM network to advance polyp detection further while maintaining real-time speed. The model is trained on custom data.

Methods

This section explains the software and hardware for our polyp detection system. We call our polyp detection system ENDOMIND-Advanced. An early, preliminary version of our detection system was experimentally tested and called ENDOMIND. Nevertheless, ENDOMIND did use an early version of YOLOv5 not involving our preprocessing, hyperparameter, optimization, and post-processing and was trained with much less data. First, we introduce our data sets used for training and testing the AI. Afterward, we illustrate typical challenges in the field of automatic polyp detection. We continue by showing our data preprocessing and data augmentation. We then show the full polyp detection system and explain its components. The full polyp detection system involves the CNN YOLOv5 and our implemented post-processing solution Real-Time-REPP, which uses an algorithm called REPP. We close this chapter by elaborating on the clinical application of our system.

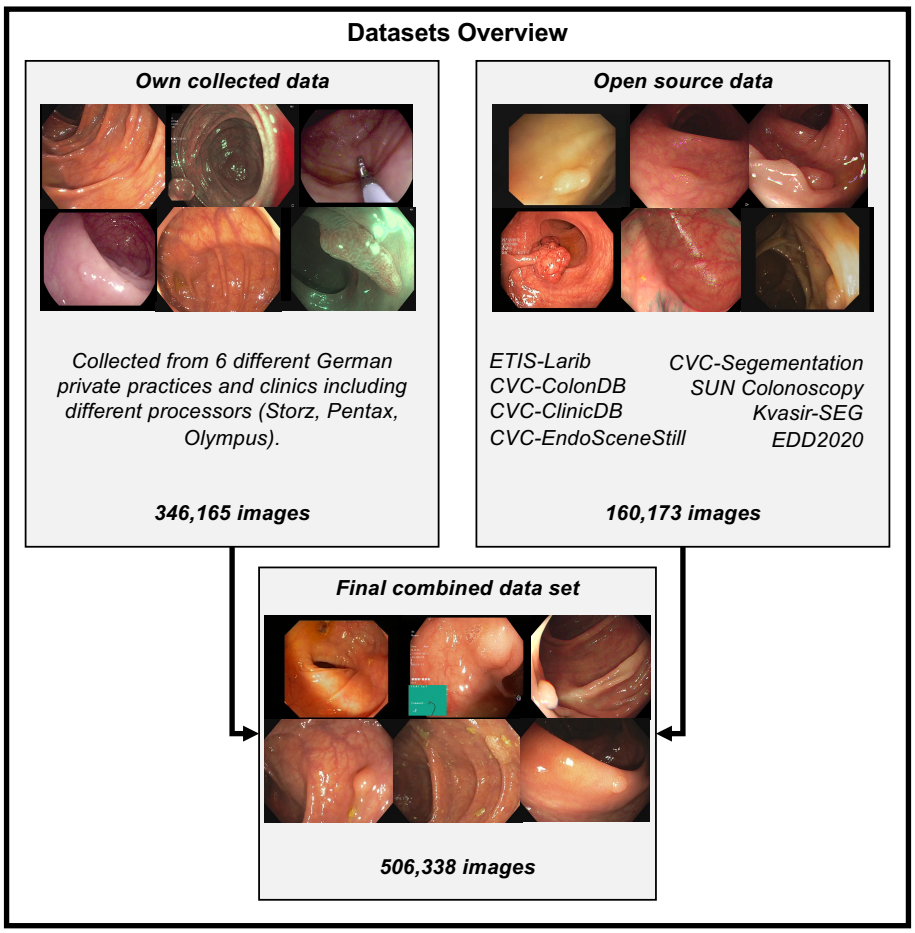


Figure 2 Data sets overview. This figure illustrates all the data we combined and gathered for training the polyp detection system. Open-source data is combined with our data collected from different German private practices to create one data set with 506,338 images. Storz, Pentax, and Olympus are different endoscope manufacturing companies. We collected the data using their endoscope processors.

Data sets

Obtaining qualitative data on an appropriate scale is often one of the biggest problems for applying deep learning methods. This is no different for colonoscopy videos/images for polyp detection. The difficulties in the acquisition are due to data protection issues on the one hand and the expensive and time-consuming, but necessary, annotation of the data by experienced medical experts. Therefore, for the development of our model, we use our own data and all the public available data we could find on the internet and in the literature. For training our model, we combined the available online sources and our own data to forge a data set of 506338 images. Figure 2 shows an overview of the data material. The details about creating our own data set will follow below. All data consists of images and bounding box coordinates of boxes referring to the image. For a listing of publicly available data sets we used, we show the following overview:

- CVC-ColonDB [52] 2012: CVC-ColonDB contains 300 still polyp images that were extracted from 15 video sequences. A random sample of 20 frames per sequence was obtained to therefore show a significantly view of the polyp. The size of the images is 348×288 pixels. The data is available on request in the CVC-Colon repository[1].

- ETIS-Larib [53] 2014: Contains 196 polyp images from 34 different videos and shows 44 different polyps. ETIS-LaribPolypDB [54] from the *MICCAI 2015 Endoscopic Vision Challenge* and was used as the testing data set in the challenge. Here we include this data set in our training data set. It has 196 polyp images with the corresponding mask for boxes. For our training we extracted the bounding boxes from the segmentation masks. The size of the images is 348×288 pixels. The data is available on request in the CVC-Colon repository[1].

- CVC-VideoClinicDB [55] 2017: The CVC-VideoClinicDB [56] data set was provided in the context of the GIANA sub-challenge that was part of the *MICCAI 2017 Endoscopic Vision Challenge*. This data set contains 18,733 frames from 18 videos without ground truth and 11,954 frames from 18 videos with ground truth. We exclusively used these frames for final evaluation. It has to be noted that the ground truth masks that labels a polyp is approximated by using ellipses. Furthermore, we also filtered out all images that had no polyps (empty mask) and only used frames with at least one polyp for training. The size of the images is 574×500 pixels. The data is available on request in the CVC-Colon repository[1].

- CVC-EndoSceneStill [57] 2017: Is a combination of *CVC-ColonDB* and *CVC-ClinicDB* and contains 912 polyp images from 44 videos of 36 patients. CVC-EndoSceneStill [58] is a data set that combines CVC-ColonDB [59] (CVC-300) and CVC-Clinic-DB [60, 61] (CVC-612). Both data sets got for each image a border, specular, lumen and segmentation mask. The border mask marks the black border that is present around each image, the specular mask indicates

[1] http://www.cvc.uab.es/CVC-Colon/index.php/databases/

the reflections that comes from the endoscope light and the lumen mask labels the intestinal lumen which is the space within an intestine. The segmentation mask contains polyp markings that tag visible polyps within a picture. Because we need the bounding box from a polyp we only used the segmentation masks and extracted a bounding box from it by calculating a box that fits around single blob. The data set CVC-ColonDB [58, 59] contains 300 selected images from 13 polyp video sequences with a resolution of 574×500 and CVC-Clinic-DB [58, 60, 61] holds 612 images from 31 polyp video sequences with a size of 348×288 pixels. The data is available on request in the CVC-Colon repository[1].

- Kvasir-SEG [62] 2020: The data set contains 1000 polyp images with corresponding 1071 masks and bounding boxes. Dimensions range from 332×487 to 1920×1072 pixels. The images were verified by gastroenterologists from *Vestre Viken Health Trust* in Norway. Most of the images have general information displayed on the left side and some have a black box in the lower left corner which covers information from the endoscope position marking probe created by ScopeGuide (Olympus). The data is available in the Kvasir-SEG repository[2].

- SUN Colonoscopy Video Database [63] 2021: The database was developed by Mori Laboratory, Graduate School of Informatics, Nagoya University. It contains 49,136 fully annotated polyp frames taken from 100 different polyps. These images were collected at the Showa University Northern Yokohama and annotated by the expert endoscopists at Showa University. Also, 109,554 non-polyp frames are included. The size of the images is 1240×1080 pixels. The data is available in the SUN Colonoscopy Video repository[3].

- CVC-Segementation-HD [58] 2017: This data set was made available within the GIANA Polyp Segmentation sub-challenge that was part of the *MICCAI 2017 Endoscopic Vision Challenge*. It contains 56 high-resolution images with the size of 1920×1080 pixels. For each image there exists a binary mask from which we have extracted the bounding boxes. The data is available on request in the CVC-Colon repository[4].

- Endoscopy Disease Detection Challenge 2020 (EDD2020) [64]: The EDD2020 challenge released a data set containing five different classes with masks and bounding boxes for each image and polyp instance. For our task, we extracted all images that are labeled as polyp and stored the relevant bounding boxes into a custom JSON file. This data contains 127 images. The size of the images is 720×576 pixels. The data is on request in the ENDOCV repository[5].

Own data creation Previously, we designed a framework that utilizes a two-step process involving a small expert annotation part and a large non-expert annotation

[2] https://datasets.simula.no/kvasir-seg/
[3] http://sundatabase.org/
[4] http://www.cvc.uab.es/CVC-Colon/index.php/databases/
[5] https://endocv2022.grand-challenge.org/Data/

part [65]. This shifts most of the workload from the expert to a non-expert while still maintaining proficient high-quality data. Both tasks are combined with AI to enhance the annotation process efficiency further. Therefore, we used the software Fast Colonoscopy Annotation Tool (FastCat) to handle the entirety of this annotation process. This tool assists in the annotation process in endoscopic videos. The design of this tool lets us label coloscopic videos 20 times faster than traditional labeling. The annotation process is split between at least two people. At first, an expert reviews the video and annotates a few video frames to verify the object's annotations. In a second step, a non-expert has visual confirmation of the given object and can annotate all following and preceding images with AI assistance. To annotate individual frames, all frames of the video must be extracted. Relevant scenes can be pre-selected by an automated system. This prevents the expert from reviewing the entire video every single time. After the expert has finished, relevant frames will be selected and passed on to an AI model. This allows the AI model to detect and mark the desired object on all following and preceding frames with an annotation. The non-expert can adjust and modify the AI predictions and export the results, which can then be used to train the AI model. Furthermore, the expert annotates the Paris classification [66], the size of the polyp, and its location, as well as the start and end frame of the polyp and one box for the non-expert annotators.

We built a team of advanced gastroenterologists and medical assistance. We created a data set of 506338 images, including the open-source images listed above. Figure 2 shows an overview of the different data sets. Our data set consists of 361 polyp sequences and 312 non-polyp sequences. The polyp sequence was selected in high quality as we were generally only annotating the first 1-3 seconds of the polyp's appearance, which is most critical for detecting polyps in a real clinical scenario. We combined training material from six centers involving three different endoscope manufacturers named Karl Storz GmbH und Co. KG (Storz), Ricoh Company Ltd. (Pentax), and Olympus K.K. (Olympus). We create a data set of 24 polyp sequences involving 12161 images and 24 non-polyp sequences involving 10695 images for the test data (EndoData). Therefore, the test data consists of additional 22856 images.

Current Challenges

There are still some challenges left. The most important of these can be divided into two categories: on the one hand, the hurdles to achieving the actual goal, real-time support of physicians, and on the other hand, problems arising from the acquisition or use of the data sets. The volume and quality of the data is a constant problem factor, and although there are various ways to deal with these problems, they do not represent an optimal solution.

Training and testing data sets The biggest problem faced by most papers, e.g., [67] or [68], is the low availability of usable data sets. This refers not only to the number and size of data sets but also to the usually significant imbalance between the two classes *healthy frames* and *frames with polyps*. The lack of availability of pathological data is a common problem for medical deep learning applications. However, there are also various attempts to deal with it.

One widely used approach by Kang et al. [67] is *transfer learning*, which uses a model that has already been pre-trained on a non-medical data set and is re-trained with a polyp data set. The advantage is that sufficiently large non-medical data sets are readily available. With these, general problem-independent skills like edge detection can already be learned well and then only need to be fine-tuned to the specialized application.

Another method that almost all papers use as well is *Data Augmentation* of the training data. This involves slightly modifying the existing training data using various methods, thus increasing both the amount of data available and the system's robustness to the applied transformations. Examples of such transformations are rotation, mirroring, blur, and color adjustments [19].

An interesting approach by Guo et al. [69] is that the test data are also augmented at test time. More precisely, they are rotated and then passed to the model. To arrive at the result for the original image, all generated masks are rotated back accordingly and averaged. This makes the system more robust against rotation and leads to better accuracy.

Other ideas can be found in Thomaz et al. [70], where a CNN is used to insert polyps into actually healthy images to increase the available training data. In Qadir et al. [71], a framework is created to annotate data that can generate the rest of the segmentation masks with only a few ground truth masks.

Video artifacts A problem that is often still little considered is the influence of video artifacts, such as reflections or blurring, on the detection rate of the methods. Attempts have been made to detect these and use artifact-specific methods to restore the frames; for example, the *Endoscopy artifact detection (EAD 2019) challenge* [72] was also conducted last year for this purpose.

The article by Soberanis-Mukul et al, [73] examines in detail the impact of artifacts on polyp detection rates. This allowed us to determine the artifact types that had the most significant influence. With these, a multi-class model was developed to recognize the different artifact types and the polyps. Since the artifacts were known, regions affected by artifacts could be avoided to be wrongly classified as polyps and polyps containing artifacts could be better classified.

Real-time application To support a physician in a real-world scenario, models should be real-time capable, meaning they should achieve a processing speed of about 25 FPS, as colonoscopy videos usually run at 25-30 FPS. Newer systems may run with up to 50-60 FPS. Of course, some speedup can be achieved by using appropriate hardware. But concerning real-world use, speed measurement should be performed on hardware that is reasonable for a physician or hospital.

Data-preprocessing

To ensure high processing speed while maintaining high detection accuracy, we rescale the images to a size of 640x640 pixel. This rescaling allows the detection system to be efficient and high performing, maintaining a speed of 20ms on an NVIDIA RTX 3080 GPU. In the clinical application subsection, we further explain the use of different GPUs and the GPU requirements for a system capable of real-time

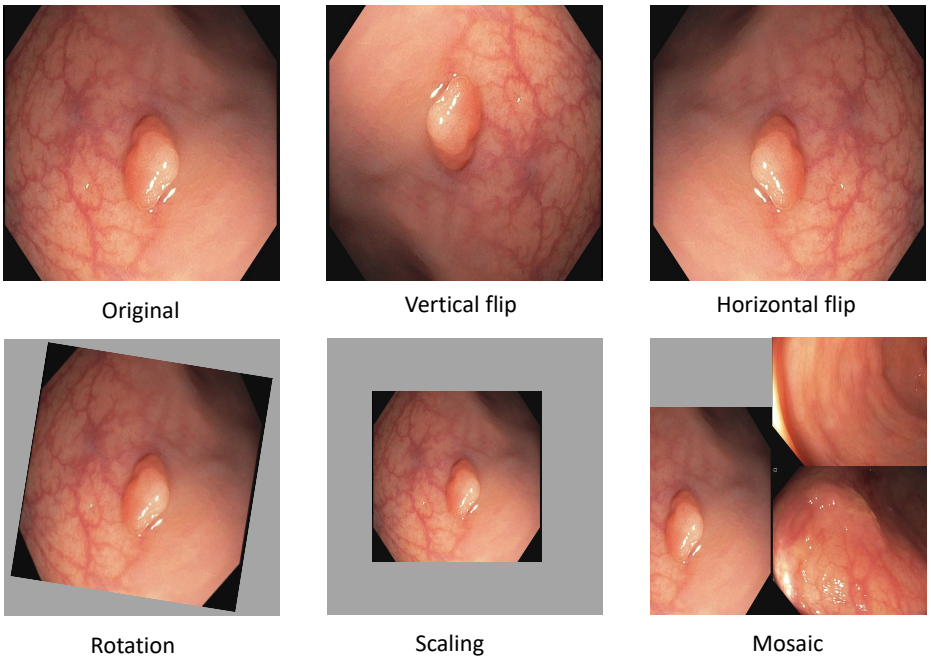


Figure 3 Data augmentation for polyp detection. This figure shows the isolated augmentation we perform to create new training samples. All of these are executed together with a certain probability in our implementation.

processing. Additionally, we transfer the image and model to half-precision binary floating-point (FP16). Normally most machine learning models are in precision binary floating-point (FP32). With FP16 the model calculates faster but maintains high performing results. The next step is image normalization. All image pixels are normalized in the following way: The min-max normalization function linearly scales each feature to the interval between 0 and 1. Rescaling to the interval 0 and 1 is done by shifting the values of each feature so that the minimum value is 0. Then, a division by the new maximum value is performed (which gives the difference between the original maximum and minimum values).

The values in the column are transformed using the following formula:

$$X_{sc} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

After normalization, we apply data augmentation. In the field of deep learning, augmenting image data means using various processes to modify the original image data. We are applying the augmentation displayed in figure 3. The most basic augmentation we apply is the flip augmentation. This is well suited for polyps as the endoscope can be easily rotated during colonoscopy. Here, the image is flipped horizontal way, vertical, or both. We applied a probability of 0.3 for up and down flips and a vertical flipping probability of 0.5. We additionally apply rescaling to the image with a probability of 0.638. Rescaling creates polyps in different sizes and therefore adds additional data to our data set. The translation moves the image

along the horizontal axis. Furthermore, we applied a low probability of 0.1 to rotate the image with a randomly created degree. For example, 20-degree rotation clockwise. As the last augmentation, we apply mosaic data augmentation. Mosaic data augmentation mixes four images into one image. Thereby, the image is rescaled, causing the images to appear in a different context. Mosaic data augmentation is applied with a probability of 0.944. These data augmentations are only applied to the training data.

Polyp-detection-system

As illustrated in figure 4, the polyp detection system starts with an input of a polyp image sequence. A polyp image sequence consists of a stream of single images extracted from a grabber of the endoscope in real-time. t states the currently processed frame. $t-1$ denotes the frame before t, $t-2$ the frame before $t-1$, etc. ws denotes our new parameter window size, which we introduce to apply Real-Time Robust and efficient PostProcessing (RT-REPP). The polyp image sequence is now passed to the polyp detection system. The polyp detection system consists of two parts: The CNN detection architecture, here YOLOv5, and the post-processing, here Real-Time-REPP (RT-REPP). The trained Yolov5 algorithm is now predicting boxes and passing those boxes to RT-REPP. RT consist of 3 main steps: First, boxes are linked across time steps, i.e. frames. This step is linking boxes according to the linking score. Details on the linking score are displayed in a subsection below in chapter about REPP. Second, unmatched boxes or boxes which do not meet specific linking and prediction thresholds are discarded through the system. Third, the boxes are adjusted by using the predicted boxes from past detections. Finally, the filtered detections are calculated and displayed on screen.

YOLOv5

In an end-to-end differentiable network, the YOLOv5[6] model was the first object detector to connect the technique of predicting bounding boxes with class labels. The YOLOv5 Network consists of three main pieces. The neck is a construction of multiple layers used to mix and combine image features to pass the prediction forward. The head takes the features from the neck and tries to predict boxes and classes. They use a convolutional neural network that aggregates and forms image features at different granularities for the backbone. To create the bounding boxes, YOLOv5 predicts them as deviations from several anchor box dimensions. In figure 5 we illustrate the YOLOv5 architecture.

CSP as Backbone Cross Stage Partial Network (CSP) model is based on DenseNet, which was created to connect layers in CNNs (convolutional neural networks) to solve the vanishing gradient problem. Yolov5 created CSPDarknet as the backbone by incorporating a cross-stage partial network (CSPNet) into Darknet. The most significant change of CSPDarknet is that the DenseNet has been reworked to divide the base layer's feature map by cloning it and sending one copy via the dense block while sending the other directly to the next stage. Therefore, the CSPDarknet solves the Problem of vanishing gradients in large-scale backbones. This is accomplished

[6] https://github.com/ultralytics/yolov5

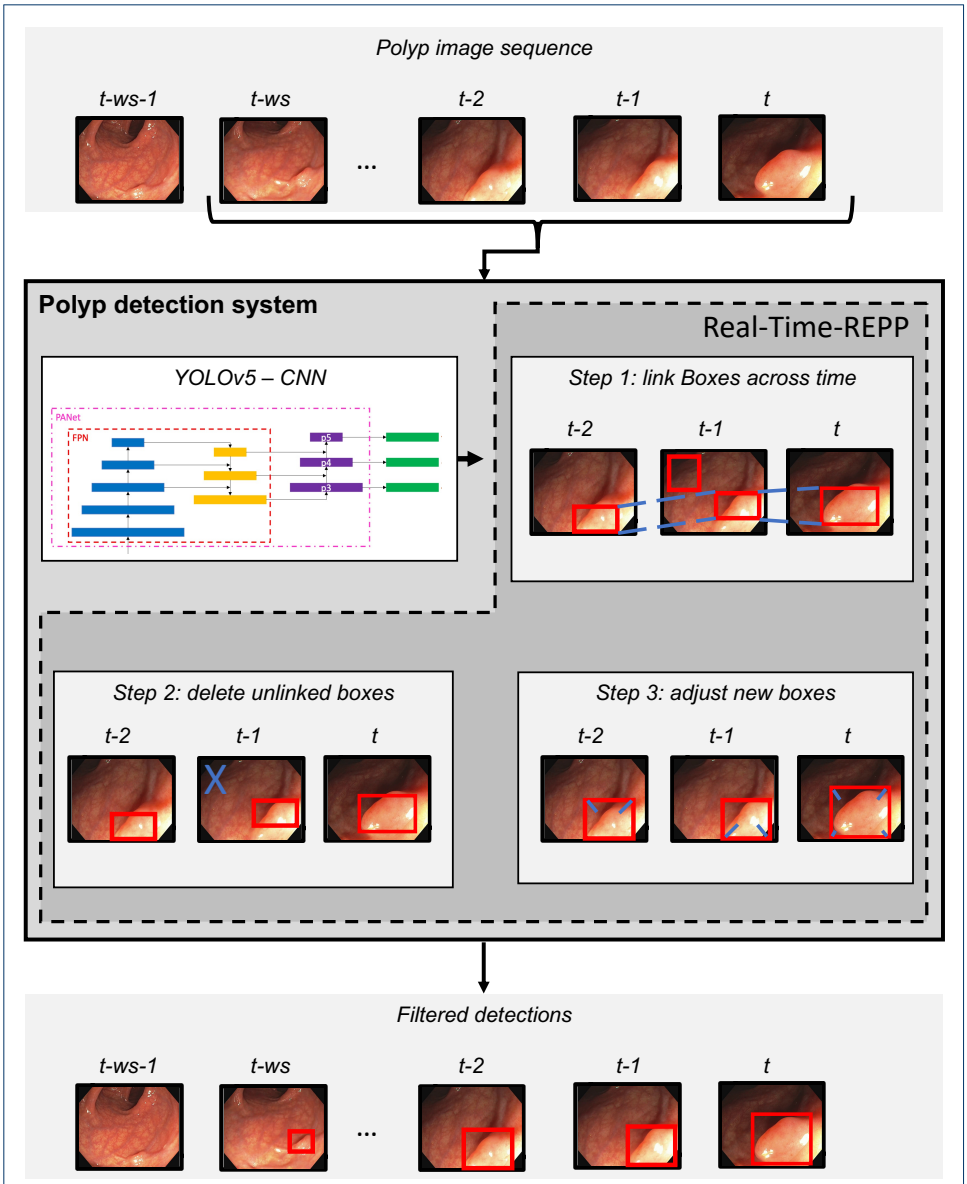


Figure 4 Overview of the polyp detection system. This figure shows all the steps of the whole polyp detection system. The start is an input of a polyp sequence ending with the last frame from the endoscope (t). From this sequence, ws frames are extracted and given to CNN architecture. Then detections are performed with YOLOv5, and the predicted boxes are post-processed by RT-REPP. Afterward, final filtered detections are calculated.

by splitting the base layer's feature map into two sections and combining them using a suggested cross-stage hierarchy. The fundamental idea is to separate the gradient flow to propagate over several network pathways. By varying concatenation and transition phases, it was demonstrated that the propagated gradient information could have a considerable correlation difference. In addition, CSPNet may significantly minimize the amount of processing required and enhance inference speed and accuracy. CSPDarknet uses two BottleNeckCSPs and one Spatial Pyramid Polling (SSP) shown in figure 5. SPP is a pooling layer that removes the network's fixed size limitation, allowing a CNN to operate with changing input sizes. It aggregates the

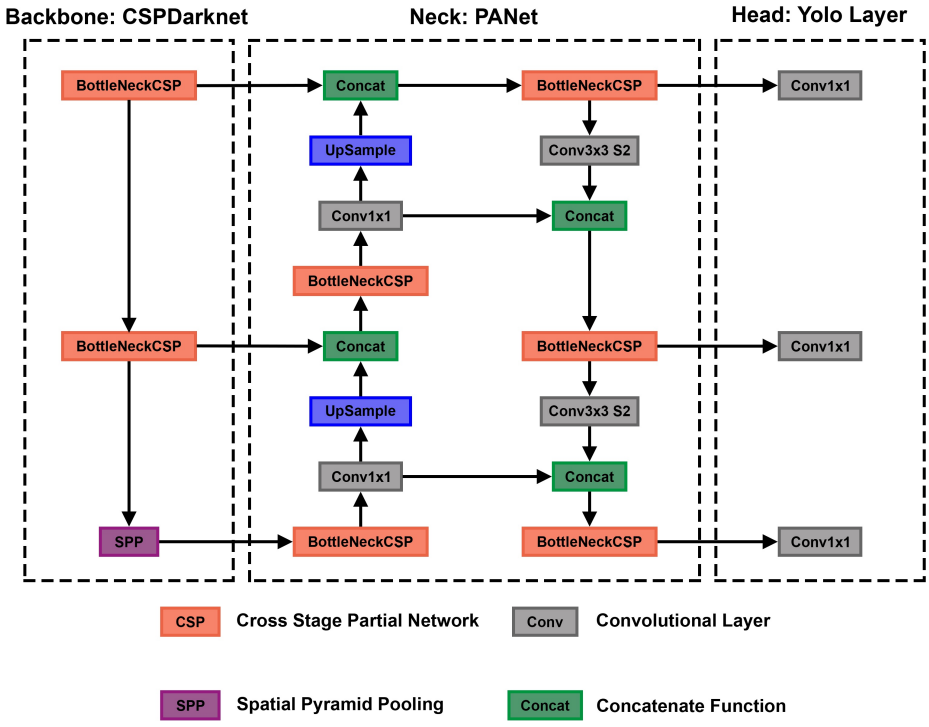


Figure 5 Detailed overview of the YOLOv5 arichtecture. This overview shows the whole architecture of YOLOv5. The starting point is the backbone CSPDarknet which is the main feature extractor. These extracted features are then given to the PANet neck structure at three stages. Finally, in the Head three outputs are given. These three outputs are specially designed for small, medium and large objects. This figure is adapted from Xu et al. [74].

features and provides fixed-length outputs that are then sent into the next layer or classifier. This works by pooling the results of each spatial bin (like max-pooling). The SSP produces kM-dimensional vectors, with M being the number of bins and k the number of filters in the last convolutional layer. Therefore, the output is a fixed-dimensional vector.

PA-Net as Neck The YOLOv5 architecture uses a path aggregation network called PANet as its neck to improve the information flow. Figure 6 illustrates the PANet and its connections with the whole architecture in more detail. PANet uses a novel feature pyramid network (FPN) topology with an improved bottom-up approach to improving low-level feature propagation. In the present architecture, the path starts with the output of the SPP from the backbone, which is passed to a CSP. This output is sent into a convolutional layer and is then upsampled. The result is then concatenated with the output from the second CSP in the backbone through a lateral connection and passed through the same combination again, which is then concatenated with the output from the first CSP of the backbone. Simultaneously, adaptive feature pooling is employed to restore broken information paths between each proposal and all feature levels. It is a fundamental component that aggregates features from all feature levels for each proposal, preventing outcomes from being allocated randomly. Furthermore, PANet uses Fully-Connected Fusion. This

augments mask prediction with small fully connected (fc) layers, which have complementary features to the FCN initially utilized by Mask R-CNN, to capture multiple perspectives on each proposal. Information diversity increases and higher quality masks are generated by combining predictions from these two perspectives. Both object detection and instance segmentation share the first two components, resulting in a much-improved performance for both tasks.

The implementation of adaptive feature pooling is straightforward. First, they map each suggestion to distinct feature levels. Next a function is utilized to pool feature grids from each level, following Mask R-CNN. The feature grids from different levels are then fused using a fusion operation (element-wise max or sum). To enable the network to adapt features, pooled feature grids are passed through one parameter layer individually in the following sub-networks, followed by the fusion operation. The box branch in FPN, for example, contains two fc levels. After the first layer, they do a fusion process. They position the fusion operation between the first and second convolutional layers since Mask R-CNN uses four consecutive convolutional layers in its mask prediction branch. For further prediction, such as classification, box regression, and mask prediction, the fused feature grid is utilized as the feature grid of each proposal. Instead of fusing information from multiple feature maps of the input picture pyramid, their method focuses on fusing information from the network feature hierarchy.

The primary route is a tiny FCN with four convolutional layers in a row and one deconvolutional layer. Each convolutional layer has 256x3x3 filters, whereas the deconvolutional layer upsamples by two. Like Mask R-CNN, it predicts a binary pixel-wise mask for each class individually to decouple segmentation and classification. A short path from layer conv3 to a fc layer is also created. There are two 33 convolutional layers, the second of which reduces the computational cost by halving channels. To forecast a class-agnostic foreground/background mask, a fc layer is employed. It is not only efficient, but it also allows the fc layer's parameters to be trained with more data, resulting in improved generality. They employ a mask size of 28x28 such that the fc layer generates a 784x1x1 vector. The mask predicted by FCN is reshaped to the same spatial size as this vector. The final mask prediction is obtained by combining the masks of each class from FCN with the foreground/background prediction from fc. Compressing the concealed spatial feature map into a short feature vector, which loses spatial information, is avoided by using just one fc layer for final prediction instead of several. Finally, the YOLOv5 head, the YOLOv5 layer, creates three different feature maps to provide multi-scale prediction, allowing the model to handle small, medium and large objects.

Hyperparameter optimization To further develop the YOLOv5 architecture for the polyp detection task. We use a genetic algorithm to find optimal hyperparameters further. This algorithm tries to gain the optimal solution of the objective function by selecting the best or fittest solution. Therefore, random mutations are used to try our new hyperparameters. We define our fitness function to be the highest possible F1 score. Therefore, the genetic algorithm is using its default set of hyperparameters, training the entire YOLOv5 algorithm in our case till 15 epochs, then calculating the F1. After that, the hyperparameters are mutated and training is restarted. If

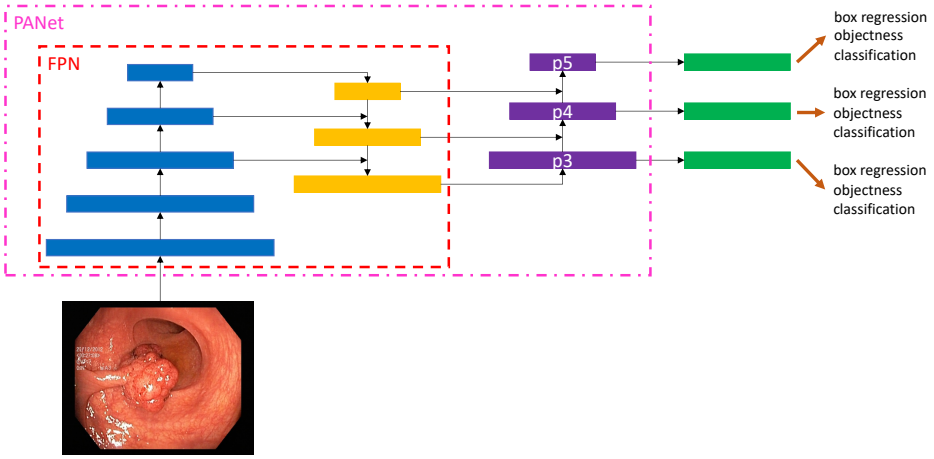


Figure 6 Overview of the PANet of YOLOv5. This overview shows a more detailed view of the PANet structure in YOLOv5. The starting point is a polyp input image. Here the FPN feature pyramid architecture is illustrated in interaction with the PANet. Finally, three outputs are given. These three outputs are specially designed for small (p5), medium (p4) and large (p3) objects.

the calculated F1 score is higher than the scores in the past, the best score and its hyperparameters are saved. After iterating over 10.000 genetic optimizations, we found our final parameters and retrained the whole algorithm for 54 epochs with the found hyperparameters. At this point, the algorithm is stopped through early stopping.

Robust and Efficient Post-Processing (REPP) and Real-Time REPP (RT-REPP)

An object detector processes each frame individually and outputs a set of detections. For an incoming stream of frames this implies that each frame is viewed independently of the previous and subsequent frames. As a result, information is lost and the performance of such detectors can significantly differ between images and video. Moreover, video data confronts the object detector with unique challenges like blur, occlusion, or rare object poses. To improve the results of the object detector for video data, the post-processing method REPP [11] relates detections to other detections among consecutive frames. Thus the temporal dimension of a video is also included. The basic idea of REPP is to link detections across consecutive frames by evaluating their similarities and refining their classification and location. This will help to suppress and minimize false-positives detections. The algorithm can be divided into three modules: (1) Object detection, (2) Detection linking, (3) Detections refinement. Figure 7 shows an overview of the REPP modules.

Object Detection It works on any object detector that provides bounding boxes and a class confidence score. For each frame t the detector delivers a set of object detections. Each detection o_t^i is described by its location and geometry $bb_t^i = \{x, y, w, h\}$, semantic information like the vector of class confidences $cc_t^i \in \mathbb{R}^C$ with C for the number of classes and the appearance of the patch $app_t^i \in \mathbb{R}^{256}$.

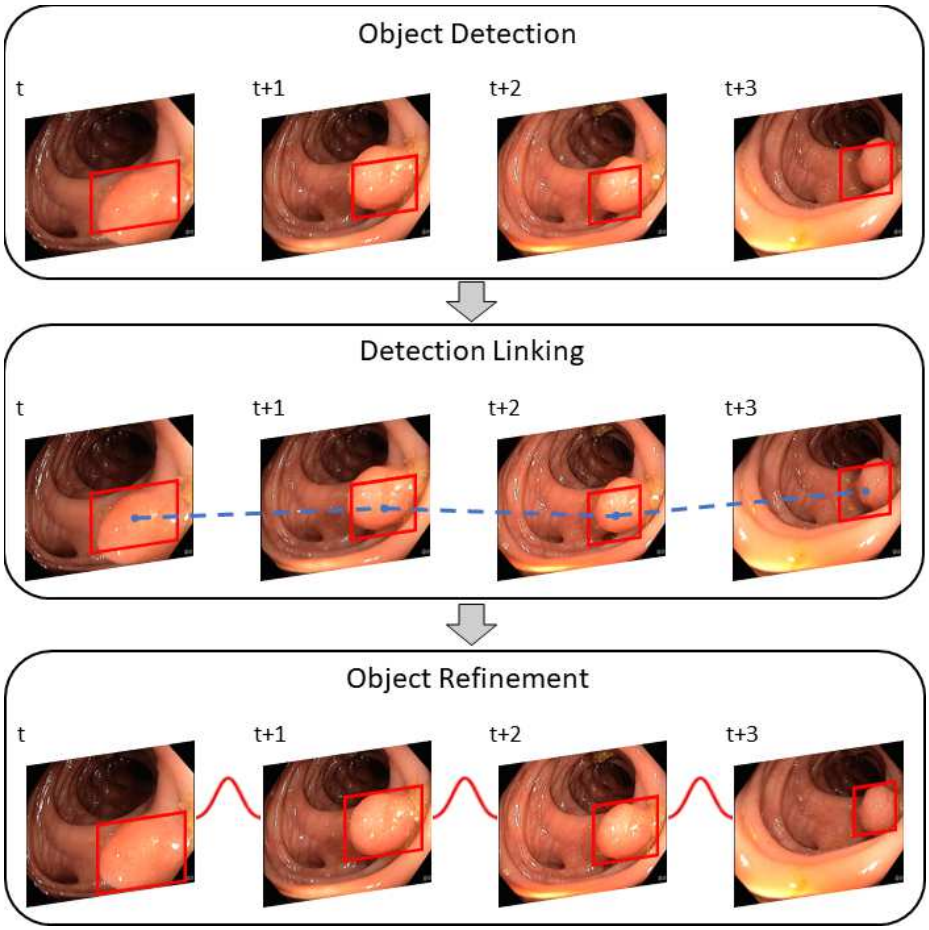


Figure 7 REPP modules for video object detection post-processing. Object detector predicts polpy for a sequence of frames (a). Then link all bounding boxes across frames with the help of the defined similarity (b). Lastly, refine detections to minimize false-positives. This figure is adapted from Sabater et al. [11].

Detections Linking Linking detections along the video by creating a set of tubelets as long as corresponding objects are found in the following frames. To link two detections between two consecutive frames a similarity function is used.

$$f_{\text{loc}} = \{\text{IoU}, d_{\text{centers}}\} \tag{1}$$

$$f_{\text{geo}} = \{\text{ratio}_w, \text{ratio}_h\} \tag{2}$$

$$f_{\text{app}} = d_{\text{app}} \tag{3}$$

$$f_{\text{sem}} = f^a_{\text{sem}} \cdot f^b_{\text{sem}} \tag{4}$$

Where f_{loc} is the location which is specified through the Intersection over Union (IoU) and the relative euclidean distance between two bounding box center points (d_{center}). f_{geo} is the geometry of the bounding boxes which is defined as the ratio of width (ratio_w) and height (ratio_h) between the two bounding boxes. f_{app} is the appearance similarity in which the Euclidean distance between the appearance em-

beddings (d_{app}) are calculated. Lastly, f_{sem} is the dot product of the class confidence vectors. Using these features, a link score (LS) is calculated.

$$LS(o_t^i, o_{t+1}^j) = f_{\text{sem}}\mathbf{X}(f_{\text{loc}}, f_{\text{geo}}, f_{\text{app}}) \tag{5}$$

Algorithm 1 shows a general description to get pairs of frames. The algorithm goes through each frame, calculates the distance between each object in both frames, and saves it in a distance matrix. The objects with the lowest distance are then considered a pair.

Algorithm 1 Get a list of pairs of frames that are linked accross frames

```
function getPairs(predictions)
    for index ← 0 to count of frames do
        predictionsFrame1 ← predictions[index]          ▷ Get frame predictions from current index
        predictionsFrame2 ← precditions[index + 1]                    ▷ Predictions of next frame
        framePairs ← empty list
        if length(predictionsFrame1) ≠ 0 and length(predictionsFrame2) ≠ 0 then
            distances ← 2D-Array with 0 for each cell
            for i ← 0 to length(predictionsFrame1) do
                for j ← 0 to length(predictionsFrame2) do
                    distances[i][j] ← LogReg(predictionsFrame1[i], predictionsFrame2[j])
                end for
            end for
            framePairs ← solveDistances(distances)          ▷ Extract pairs from distance matrix
        end if
        pairs.append(framePairs)
    end for
    return pairs
end function
```

Next, tubelets are created (Algorithm 2) from a list of linked pairs. Tubelets link all bounding boxes that identify the same object across a series of frames.

Algorithm 2 Tubelets creation from list of linked pairs

```
function getTubelets(predictions, pairs)
    tubelet ← empty list
    for each frame do
        for each pair in following frames do
            if frame has no pair then
                start new tubelet
            end if
            if frame has pairs then
                append box from pair to tubelet
            end if
        end for
    end for
    return tubelets
end function
```

Detection refinement With the use of the tubelets the classification and location are improved. First, the detection classification scores are recalculated. Therefore, all class confidence vectors are averaged and assigned to each detection within the tubelet (see Algorithm 3). This helps the correct mislabeled detections and disambiguate those with low confidence.

The next step is to improve the detection positions. Each coordinate of a linked object is treated as noisy time series. Smoothing is used to alleviate the noise with

Algorithm 3 Rescore tubelets

```
1: function RESCORETUBELETS(tubelets)
2:     for each t ∈ tubelets do
3:         s_avg = (1/|t|) · Σ_{p∈t} s_p        ▷ Average score s of predictions p of tubelets
4:         ∀p ∈ t : s_p = s_avg                 ▷ Assign average to all prediction scores
5:     end for
6:     return tubelets
7: end function
```

the help of a one-dimensional Gaussian filter convolution along with each time series. The smoothed series are then used as the set of coordinates of the object in the tubelet (Algorithm 4 line 7).

The final REPP algorithm is simply the combination of all aforementioned algorithms executed in order. First, filter detection predictions, then get all pairs and afterward computes the tublets out of the pairs. Then rescore detections within tubelets and recoordinate them for improved prediction results. Lastly, filter all predictions that do not reach a certain threshold and convert them to a specific prediction result format (e.g. COCO format).

Algorithm 4 REPP

```
1: function REPP(objectDetectionPredictions)    ▷ Gets all predictions from detection network
2:     videoPredictions ← FILTERPREDICTIONS(objectDetectionPredictions)
3:     pairs ← GETPAIRS(videoPredictions)
4:     tubelets ← GETTUBELETS(videoPredictions, pairs)
5:     tubelets ← RESCORETUBELETS(tubelets)
6:     if recoordinate == True then              ▷ Tubelets reccordination is optional
7:         tubelets ← RECOORDINATETUBELETS(tubelets)
8:     end if
9:     predictions ← TUBELETSTOPREDICTIONS(tubelets)    ▷ Filter and convert to specific format
10:    return predictions
11: end function
```

Since REPP is a post-processing method that only works on finished videos, REPP can include past and future frames. RT-REPP can only include past frames. Therefore, REPP should always lead to better results than the RT-REPP algorithm. However, REPP is not applicable in real-time. We modified the code to use it in our real-time application calling the algorithm RT-REPP. For that, we use a buffer of pre-defined length and run the original REPP within the buffer. The size of the buffer can be changed to fit the available resources. The greater the length, the longer it takes to execute REPP. Before REPP is executed for the first time, the buffer must be complete, i.e., there is a size-dependent delay at the beginning of each video. To overcome this delay, an alternative way is to run REPP from the start of the first frame and execute REPP for every new frame until the buffer is full. This causes bigger buffers for each execution until it reaches its pre-defined maximum capacity. A full buffer is passed, and the oldest frame gets deleted when a new frame is inserted. Since the delay times for our application are relatively short, we accept this short delay. See algorithm 5 to understand our basic idea behind real-time REPP. We define buffer size or window size as ws, and ws can vary the length. We figured a ws of 300 is sufficient when using a real-time stream of frames with 25 FPS. A ws of more than 300 does not improve the accuracy significantly.

To implement RT-REPP in our real-time system, we had to combine C++ and python to function correctly. Therefore, we used the lightweight header-only library

pybind11, which allows us to use C++ types and methods in python and *vice versa*. To our knowledge, REPP and RT-REPP have never been used before in the domain of polyp detection. In figure 8 the workflow of Real-Time REPP is illustrated.

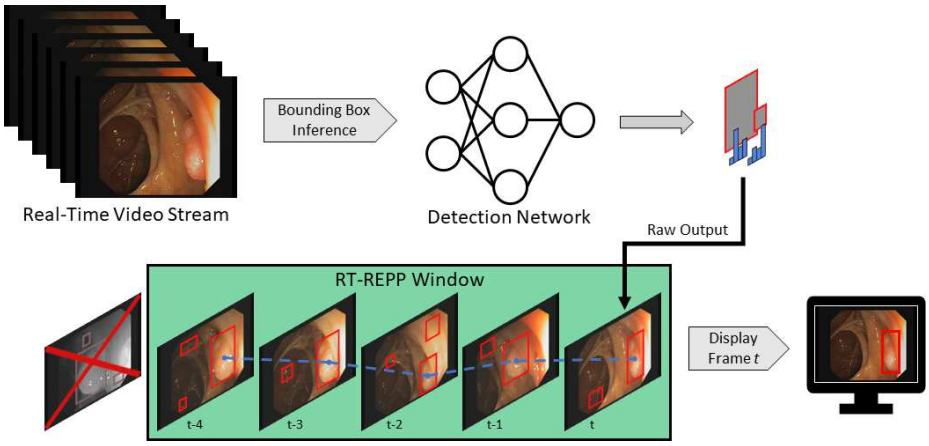


Figure 8 Real-Time REPP. It gets a stream of video frames, where each frame is forwarded into a detection network. The result of the current frame is stored into the buffer (green) and REPP is executed afterward. The improved result are then displayed.

Algorithm 5 RT-REPP

```
function RT-REPP(framePrediction)
    if buffer is full then
        delete oldest frame
    end if
    add prediction to buffer
    run REPP on buffer
end function
```

Clinical Application

To develop a system for clinical trials, it is mandatory to understand the current settings of examination rooms. Endoscopic and other medical equipments are complex devices that are widely tested and secured. Therefore, this is only a brief overview of these highly sophisticated components.

Figure 9 shows an example of medical devices used during endoscopic interventions. Figure 9a presents an endoscope composed of a flexible tube, controlled and operated by physicians during examination through several control buttons and physical force. A fish-eye camera is on the tip of this tube, combined with fiber for light to capture an RGB video stream. An endoscopy tower contains the entire endoscopic equipment and, most importantly, the camera's light source and an endoscopy processor (figure 9b). When connected to the endoscope it captures, the latter provides several functions, such as the process of the camera stream and outputting it as a regular video signal. This signal can be used to display it on a regular or medical monitor, in figure 9c. Together, these components provide physicians with real-time controlling and visual feedback during endoscopic interventions. Based on the giving setting, we have developed a prototype. Instead of connecting

(a) Endoscope (b) Endoscopy Tower (c) Monitor

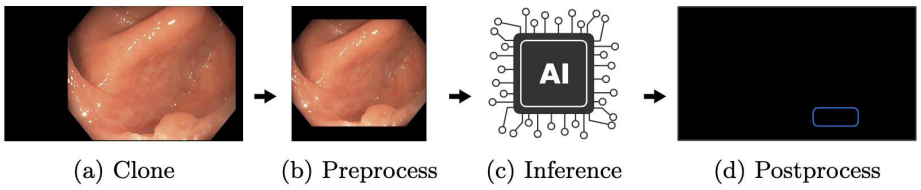
Figure 9 Setting Examination Room.

Table 3 Prototype Components

Component	Type	Info
CPU	AMD Ryzen 7 3800X	8 Cores, 3.9GHz
GPU	MSI GeForce RTX 3080 Ti	12GB GDDR6X
RAM	G.Skill RipJaws V DDR4-3200	2 × 8GB
Disk	Samsung SSD 970 EVO Plus	500GB
Mainboard	B550 Vision D	-
Frame Grabber	DeckLink Mini Recorder 4K	-

directly the endoscopy processor to the monitor, our system stands in between, capturing and processing all frames, before forwarding to the monitor.

Table 3 shows the main hardware components of our system, which opens the real-time image processing. However, these are just as important as suitable software. In order to provide physicians with the best possible user experience, all incoming frames must be displayed as fast as possible to minimize latency. To this end, image capturing, displaying, and processing are running in separate threads. The first thread makes use of the Blackmagic SDK to capture frames. This directly depends on the frame rate. For instance, Olympus CV-190 provides 50 frames per second, receiving a frame every 20ms. Therefore, it is essential to distribute the additional workload on other threads so it is not blocked. If not, incoming frames would be buffered, resulting in an overall delay across all related threads. Considering this, thread one only captures and transforms incoming data to an OpenCV matrix, passing it to subscribing pipelines.

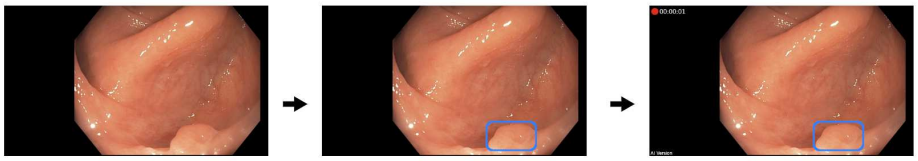


(a) Clone (b) Preprocess (c) Inference (d) Postprocess

Figure 10 AI pipeline.

One receiver is the AI pipeline in figure 10. In this thread, all incoming frames are cloned (figure 10a) to ensure that all operations on those image matrices do not interfere with other processes. On this clone, several operations, denominated

preprocessing (figure 10b), are performed. Here, frame matrices are transformed to fit the AI network. First of all, black borders are cropped. For most endoscopy processors with a resolution of 1920×1080 pixels, this results in a "closer to square" ratio since the widest black borders are on sides, while there are no or minimal black areas top and bottom. The target resolution is 640×640 pixels, so matrices are resized based on the longer edge, to fit 640 pixels, while the shorter one results in less than 640 pixels. To compensate this, without squeezing nor stretching the matrix, black padding is added to fill up to 640 pixels. In figure 10a to figure 10b this is illustrated. The resulting 640 × 640 matrix is transformed from BGR to RGB and uploaded to GPU memory. Here the matrix is then processed through YOLOv5 (figure 10c). Based on the input, it results in relative coordinates, classes, and scores for every detection. The last step is a transformation resulting in a vector of quadruples, containing xy-coordinates, width, height of bounding boxes to suit the original matrix (figure 10d). Under consideration of thresholds, detections with low confidence are removed, while remaining detections are transformed and forwarded to the display pipeline.



(a) Clone (b) Draw Boxes (c) Draw UI Elements

Figure 11 Display pipeline.

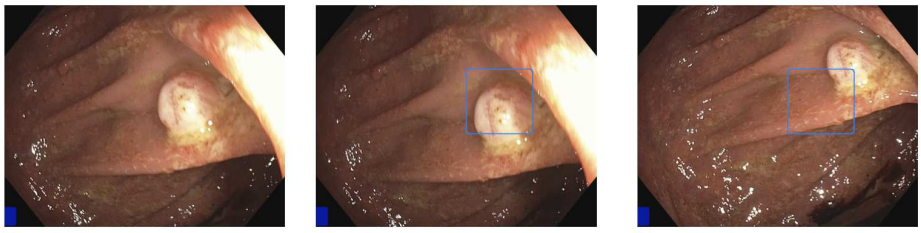
The independent display pipeline thread is designed to display captured frame matrices as fast as possible. Just like the AI pipeline, matrices are cloned at the beginning shown in figure 11. Consequently, no processing is applied on the original matrices and therefore, other pipelines remain unaffected. Then, based on the most recent detections of the AI, boxes are drawn and old ones removed. Boxes will remain on the screen until a new cycle of the AI pipeline has finished. Additionally, a few extra UI elements, such as a timer, are drawn to indicate that the AI is running before frames are forwarded and displayed. This design, as mentioned earlier, decouples AI and display pipeline. Hence, a slower AI would not directly result in higher latency for displaying. Nevertheless, the performance of the AI Pipeline is also an essential factor. As expected, faster executions lead to more inferences and, therefore, more precise boxes, given that the displayed frame is closer to the AI pipeline frame. To demonstrate this, a test was performed. To this end, under real conditions, the prototype was used on two different GPUs to show performance differences. The prototype components are listed in figure 3. Thus, a second computer streamed a colonoscopy video instead of an endoscopy processor, just like an endoscopy processor would. Meanwhile, the prototype captured the signal, as mentioned earlier. This ensured identical, reproducible conditions and guaranteed occurring polyps during the experiment. The prototype is not able to distinguish this method from a live endoscopy. Through video encoding, decoding and presentation, only minor differences could lead to insignificant frame matrices. However,

this has no effect given that this test does not evaluate AI accuracy but the latency of the entire pipeline. The streamed examination video was presented in 1920 pixels and 50fps, equivalent to streams of Olympus CV-190. In our test, we used the MSI GeForce RTX 3080 Ti, an up-to-date high-end GPU, released on June 3th 2021. Nvidia Geforce GTX 1050 Ti, a low-budget GPU two generations ago, was used for a second test run. Release date May 27th, 2016. All other hardware components and software parts were constant throughout testing.

Table 4 5000 frames system test

GPU	AI exe. count	AI avg. exe. time	AI evaluation rate
RTX 3080 Ti	2996	19.5ms	1.7 FPS
GTX 1050 Ti	313	306.7ms	16.0 FPS

In the setting of table 4 5000 frames were used. Out of those 5000 frames, the RTX 3080 Ti could execute the AI pipeline 2996 times. At the same time, the GTX 1050 Ti made 313 executions. This is based on AI avg. exe. time [7] 19.5ms and 306.7ms, respectively. It shows that under the usage of RTX 3080 Ti, there was a 15-fold performance gain. The AI pipeline was applied on every 1.7th frame on this GPU, while only every 16th frame was evaluated through the old, low-budget GPU. Based on those average numbers and a synchronized display pipeline, it would take two frames until bounding boxes are shown on display, and those would remain two more frames until they are updated again, in total four frames, 80ms, for RTX 3080 Ti. GTX 1050 Ti leads in total to 32 frames and 640ms, while 16 frames and 320ms for the first appearance of a bounding box. This does not illustrate the worst or the best-case scenario.



(a) Frame 0 (b) Box on frame 2 (c) Box on frame 16

Figure 12 Detection shift through latency.

An example has been created to show this delay for the appearance of a bounding box. Figure 12a shows a frame, which is forwarded to the AI pipeline. Since the RTX 3080 Ti needs an avg 1.7 frames, bounding boxes would be shown in frame two. This is illustrated in figure 12b. While the camera moves, frame two is slightly shifted to the bottom, but the polyp is still mainly boxed. Under the GTX 1050 Ti it takes 16 frames shown in figure 12c. The polyp is mainly outside the bounding box. A box might appear based on the speed an endoscope is pulled, even if a polyp is not displayed any longer. This is very unlikely for the RTX 3080 Ti, which in the best case shows a bounding box on the next frame. However, for slower GPUs, this should

[7] AI pipeline average execution time

be considered. On the other hand, for a slow withdrawal, it matters less. This test has been done on an actual prototype. Therefore, the software has not been altered. In addition, a video was recorded simultaneously. This is done for quality assurance and to retrieve additional test data. The recording pipeline is independent, but the GPU is used for H264 video encoding, and this causes an additional load, which might affect the performance of the AI pipeline. In general, our prototype is not designed for a specific GPU, all Nvidia GPUs with Cuda compatibility of the last five years can be used, but it will affect the user experience. In an actual examination, prototypes have been used with a MSI GeForce RTX 2080 SUPER Ventus XS OC with no significant change in user experience.

Results

For the evaluation we use two data sets. First the CVC-VideoClinicDB data set is the first benchmark data set for polyp detection in videos. Compared to early benchmarks e.g. data sets ETIS-Larib and CVC-ColonDB which just allow a comparison based on still images. Therefore, the CVC-VideoClinicDB data set has the ability to evaluate models in a more realistic scenario. As in real polyp detection the input are not images but a stream of images provided in real-time. As our architecture explained in methods is only applicable on videos or a stream of images we state the CVC-VideoClinicDB data set to be our main data set for the comparison of our methods against methods in the literature. Secondly, our own test data: In the CVC-VideoClinicDB data set, the polyp sequence begins with the polyp already in view in the first frame. Since our data set contains the entire footage, the polyps appear further into the image sequence. Hence, our own data set emulates the clinical practice more closely, which makes the evaluation even more realistic for real world application. Therefore, we can additionally calculate a metric measuring the time taken to detect a polyp. We published the code for application and evaluation of our system [8].

The quality evaluation is done via the F1-score. The F1-score describes the harmonic mean of precision and recall as show in following equations:

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN}$$

We count an annotation as true positive (TP) if the boxes of our prediction and the boxes from the CVC-VideoClinicDB data set ground truth overlap at least 50%. Additionally, we choose the mAP, which is a standard metric in object detection [75]. The mAP is calculated by the integral of the area under the precision-recall curve. Thereby, all predicted boxen are first ranked by their confidence value given by the polyp detection system. Then we computed precision and recall for different thresholds of these confidence values. When reducing the confidence threshold recall increases and precision decreases. This results in a precision-recall curve. Finally, for this precision-recall curve, the area under the curve is measured. This results

[8] https://fex.ukw.de/public/download-shares/d8NVHA2noCiv7hXffGPDEaRfjG4vf0Tg

in the mAP. Furthermore, our approach introduces new parameters to the polyp detection system. One of the parameters is the width of the detection window ws.

We created the following evaluation on our data set (EndoData). Our evaluation involves two baseline models: the YOLOv5 (base) model and the Faster R-CNN baseline. First, the YOLOv5 (base) is the basic YOLOv5 model without any hyperparameter optimization, data augmentation, post-processing, or other tweaks for polyp detection, and just the plain model trained from scratch with our data. The second baseline model is a Faster R-CNN with a ResNet-101 backbone. This involves training a Faster RCNN with default parameters using the Detectron2 framework [76].

Furthermore, we show three different stages of our polyp detection system. First, YOLOv5 advanced (adv), which is training YOLOv5 algorithm but with all our in chapter methods explained features and optimization to specialize it for the polyp detection task. Second, REPP is training YOLOv5 (adv) and adding the REPP post-processing. This is not applicable in real-time as the REPP algorithm is just applicable to a full video. Then RT-REPP, which is our version of REPP making REPP work in real-time. Our polyp detection system ENDOMIND-Advanced is in the following evaluation refered as RT-REPP. All algorithms are trained on our training data using four Quadro RTX 8000 Nvidia graphics cards. The test application is made on an Nvidia GeForce RTX 3080. The results of these models are shown in detail in tables 5, 6, 8, 9, 10.

CVC-VideoClinicDB data evaluation

To compare our polyp detection system to the published research, we use the publicly available CVC-VideoClinicDB data. To our knowledge the best performing algorithm on the data set was publish by Qadir et al. [48]. Qadir et al. [48] is therefore included into the evaluation in table 5. Table 5 shows the comparison of different baseline and advanced stage models. All values are calculated according to the CVC-VideoClinicDB challenge norm. Therefore, the means are micro means taken over all the images in the data set. We use this micro mean structure through the whole paper. Therefore, all means are micro averages over all images. Because of a mistake in the ground truth in the CVC-VideoClinicDB data set. We excluded Video 18 as 77 of 381 are labeled incorrectly.

For the F1-Score, REPP has the highest F1 and thereby is the best model. Nevertheless, REPP is not applicable in real-time as it can be calculated by combining past, present, and future predicted boxes. Therefore, REPP can just be used on finished recorded videos. Nevertheless, we like to include it in our comparison to show the enhancement using the full algorithm. The second-best model in F1 score is RT-REPP, so the real-time application we implemented to us repp in a real-time scenario. The gap from using RT-REPP vs. YOLOv5 advanced results in an improvement of 4.15% F1 score. The baseline models F-RCNN and YOLOv5 baseline are lower than those values.

Overall our results show that using our hyperparameter, data augmentation and training set up, F1 and mAP can be increased by 6.05% and 4.78%. By leveraging our implementation RT-REPP results in another improvement of 4.15% and 3.14%. REPP and RT-REPP do just attain a minimal speed reduction, resulting in roughly

Table 5 Evaluation CVC-VideoClinicDB data set. This table shows our comparison of six different polyp detection approaches on the benchmarking data CVC-VideoClinicDB. The first two models are baseline models, and the third is the best model we could find in the literature. The last three models are different stages of our polyp detection system. Precision, Recall, F1, and mAP are given in %, and the speed is given in FPS.

	Precision	Recall	F1	mAP	Speed	RT capable
YOLOv5 (base)	92.15	69.98	79.55	73.21	44	yes
Faster-RCNN	93.84	74.79	83.24	79.78	15	no
Qadir et al. [48]	87.51	81.58	84.44	-	15	no
YOLOv5 (adv)	98.53	76.44	86.09	77.99	**44**	yes
REPP	**99.71**	**87.05**	**92.95**	**86.98**	42	no
RT-REPP	99.06	82.86	90.24	83.15	43	yes

Table 6 Detailed evaluation CVC-VideoClinicDB data set. This table shows our comparison of five different polyp detection approaches on the benchmarking data CVC-VideoClinicDB. The first two models are baseline models. The last three models are different stages of our polyp detection system. F1, and mAP are given in %.

Video	YOLOv5 (base)		F-RCNN		YOLOv5 (adv)		REPP		RT-REPP	
	mAP	F1	mAP	F1	mAP	F1	mAP	F1	mAP	F1
1	78.22	87.41	92.56	88.14	85.17	91.47	94.56	97.44	89.38	94.18
2	87.35	91.87	89.48	89.19	94.62	96.91	97.48	98.48	96.48	97.96
3	75.58	80.09	81.48	77.71	80.18	84.42	86.48	87.64	82.65	85.01
4	90.04	92.16	93.35	90.39	98.00	98.99	98.35	99.50	98.29	98.99
5	76.29	82.53	78.01	85.85	78.40	87.64	83.01	90.71	78.88	88.27
6	86.23	88.59	87.05	89.42	90.07	94.83	92.05	95.43	88.41	92.83
7	60.75	67.15	69.56	78.38	66.23	76.15	74.56	85.71	71.95	82.35
8	53.93	69.52	77.22	82.65	59.16	73.66	82.22	90.11	82.22	90.11
9	74.27	77.29	84.10	87.21	76.50	87.01	89.10	94.18	85.15	91.89
10	75.28	77.36	86.33	86.00	78.22	87.25	91.33	95.29	86.61	92.61
11	90.17	92.19	94.19	94.92	95.41	97.44	99.19	99.50	98.65	99.50
12	30.81	46.22	42.51	60.09	36.78	54.01	47.51	64.86	39.85	57.14
13	84.48	89.48	84.68	87.06	89.37	94.29	89.68	93.83	90.00	94.74
14	74.35	80.49	82.20	86.42	79.09	87.88	87.20	93.05	82.20	90.11
15	48.88	62.62	52.51	66.56	52.18	69.04	57.51	73.15	55.65	71.79
16	89.45	92.97	93.63	90.32	94.54	97.44	98.63	99.50	98.36	98.99
17	52.25	64.61	56.29	68.15	57.77	72.59	61.29	75.78	49.80	65.75
Mean	73.21	79.55	79.78	83.24	77.99	86.09	86.98	92.95	83.15	90.24

Table 7 Details of the EndoData. This table shows the details of our own evaluation data (EndoData). Width and height state the size of the used frames.

Video	1	2	3	4	5	6	7	8	9	10
Frames	14947	18026	1960	1923	9277	14362	347	4627	6639	766
Polyps	2	5	1	1	2	5	1	2	4	1
Width	1920	1920	1920	1920	1920	1920	1920	1920	1920	1920
Height	1080	1080	1080	1080	1080	1080	1080	1080	1080	1080

1 FPS speed reduction for RT-REPP and 2 FPS reduction in REPP. Therefore, those algorithms can easily be added to neural networks without losing a lot of processing time.

For the detailed evaluation, we computed the mAP and F1 score for each of the 17 videos of the CVC-VideoClinicDB data set. REPP-RT detects most videos with a F1-Score of over 90%. Just videos 3,5,7,12,15,17 do have a lower score than 90%. These are also the videos in which test results are not that promising. Especially video 12 with a score of 57.14%, video 17 with a score of 65.72% and video 15 with a score of 71.79%. Therefore, in our discussion section, we will analyze those videos in more detail. The YOLOv5 base lane model also gets inferior results with a value of 46.22%. So a detection value lower than 50%.

Table 8 Evaluation of EndoData. This table shows our comparison of five different polyp detection approaches on our own data set (EndoData). The first two models are baseline models. The last three models are different stages of our polyp detection system. F1, and mAP are given in %.

	Precision	Recall	F1	mAP	Speed	RT capable
YOLOv5 (base)	78.39	80.54	79.45	77.09	44	yes
Faster-RCNN	81.85	86.20	83.97	81.74	15	no
YOLOv5 (adv)	86.21	86.43	86.32	82.28	44	yes
REPP	**90.63**	**89.32**	**89.97**	**87.24**	42	no
RT-REPP	88.11	87.83	87.97	84.29	43	yes

Table 9 Time to first detection on our own data set (EndoData). This table shows our comparison of five different polyp detection approaches on EndoData with our new metric time to first detection (FDT). The first two models are baseline models. The last three models are different stages of our polyp detection system. FDT is measured in seconds. FP denotes the number of false positives in the video.

Video	YOLOv5 (base)		F-RCNN		YOLOv5 (adv)		REPP		RT-REPP	
	FDT	FP	FDT	FP	FDT	FP	FDT	FP	FDT	FP
1	0.07	201	0.00	159	0.00	155	0.00	109	0.00	150
2	0.68	13	0.62	11	0.51	4	0.51	8	0.51	5
3	0.10	21	0.00	17	0.00	30	0.00	12	0.00	13
4	0.00	234	0.00	198	0.00	145	0.00	135	0.00	123
5	1.33	663	1.07	572	0.93	425	0.93	379	0.93	352
6	0.13	35	0.07	31	0.03	127	0.03	22	0.03	68
7	5.00	50	3.40	33	2.60	51	2.67	22	2.63	28
8	0.20	99	0.08	83	0.05	152	0.05	58	0.05	50
9	0.68	41	0.32	35	0.32	83	0.32	25	0.32	115
10	0.03	22	0.00	19	0.00	15	0.00	13	0.00	9
Mean	0.82	137.9	0.56	118.7	**0.44**	113.5	**0.45**	**78.3**	**0.44**	91.3

Table 10 Detailed evaluation of EndoData. This table shows our comparison of five different polyp detection approaches on the our own data set (EndoData). The first two models are baseline models. The last three models are different stages of our polyp detection system. F1 and mAP are given in %, and the speed is given in FPS.

Video	YOLOv5 (base)		F-RCNN		YOLOv5 (adv)		REPP		RT-REPP	
	mAP	F1	mAP	F1	mAP	F1	mAP	F1	mAP	F1
1	72.77	72.69	84.23	82.26	79.25	82.23	89.84	89.43	82.98	84.26
2	86.30	86.71	86.04	90.51	89.06	94.18	92.83	95.91	90.01	94.74
3	85.65	85.71	93.10	92.88	91.20	91.50	99.10	97.99	98.51	97.00
4	70.57	73.88	82.88	78.17	76.96	79.99	85.43	85.36	83.67	83.99
5	39.45	54.84	44.23	56.79	45.84	58.98	49.60	63.98	49.28	62.40
6	90.22	90.94	94.02	92.11	96.13	96.00	98.38	97.48	96.75	97.50
7	15.12	34.89	29.13	47.81	21.66	43.40	31.72	53.33	28.41	46.39
8	91.14	86.35	96.32	92.71	96.66	94.43	99.46	98.48	98.67	97.00
9	77.49	80.87	78.48	84.72	82.61	87.44	85.11	89.29	81.61	86.59
10	88.73	87.08	88.28	89.10	91.95	94.43	95.82	96.50	92.28	94.91
Mean	77.09	79.45	81.74	83.97	82.28	86.32	87.24	89.97	84.29	87.97

EndoData evaluation

Our own validation set (EndoData) allows us to be more specific and accurate about polyp detection. Table 7 shows an overview of the videos in the data set. In our data set, a polyp goes from unseen and then appears in the sequence. Therefore, polyps are marked precisely with their first appearance. Therefore, in our data set compared to the CVC-VideoClinicDB data set, a polyp sequence might not start when the polyp is already detected. Those early seconds are very important as it is the part where the Gastroenterologists have to see and not miss the polyp. If the polyp is not detected in the early sequence, the risk of missing the polyp increases. As we like to focus on this early detection, we introduce a second metric that can just be evaluated with a data set like ours. This metric marks the seconds from first seeing the polyp to first detecting the polyp. We call it first detection time (FDT). Additionally, we compute the false positives per video.

The evaluation for FDT is shown in table 9. For the YOLOv5 (base), just video 4 does not receive a delay in detection. Nevertheless, all polyps are detected at least once with every algorithm. The FDT of YOLOv5 base is worse in all videos than the other algorithms. The Faster R-CNN algorithm does recognize the polyp in the first frame in videos 1,3,4 and 10 for YOLOv5 (adv), REPP, and RT-REPP. The FDT does not differ except for video 7. However, the difference is very low therefore just a small trade-off while reducing FPs. This is due to the YOLOv5 adv. The first detection can be as good or worse as REPP or RT-REPP does not change the initial detection. It can just later in the preprocessing remove a detection. Nevertheless, REPP or RT-REPP have not removed a detection, and therefore, the values are the same as YOLOv5 adv. Those three approaches also detect the polyps in the first frame for videos 1,3,4 and 10 like Faster R-CNN. For 9 out of the 10 videos FDT is under 1 second, and therefore, the polyp should be sufficiently detected to show the gastroenterologist its position. Nevertheless, in video 7 there is a FDT of 2.6 seconds. Such a late detection of a polyp may miss the polyp for the gastroenterologist. However, REPP and RT-REPP are reducing the number of false positives from an average of 113.5 to 78.3 and 91.3.

Furthermore, we evaluate the algorithms on our data set with the same metrics as the CVC-VideoClinicDB data set. On the EndoData data set, the results are equivalent to the predictions of the CVC-VideoClinicDB data. The mAP is, on average, consistently lower than the F1. Additionally, REPP is again the best scoring model. Again most values are over 90% F1 value for RT-REPP. It seems the data set is even more complicated than the CVC-VideoClinicDB data set as there are just five videos with F1 scores over 90%.

Explaining the model with heatmaps

This paragraph presents a methodology to generate visual explanations for deriving insight into our polyp detection systems decisions using the Grad-CAM algorithm [77]. We follow the Checklist for Artificial Intelligence in Medical Imaging (CLAIM) [78]. Nevertheless, we changed the GRAD-CAM algorithm to fit an object/polyp detection task instead of classification. Grad-CAM receive the image of the prediction of the model, the result of the prediction and the last layers of the convolutional neural network YOLOv5. YOLOv5 outputs detections for the three scales p3 (large), p4 (medium), and p5 (small), each of them with a shape of $[bsz; n_a; h; w; (5 + n_c)]$, where bsz is the batch size, n_a is the number of anchor boxes per grid cell, h is the height of the feature map, w is the width of the feature map, four box coordinates + objectness = 5, and n_c is the number of classes. Next, a concatenation and reshaping of the three scales occur, resulting in an output of shape $[bsz; n_a \times h \times w; (5 + n_c)]$ followed by data augmentation. The augmentation identifies the scales from which the detections originate.

After that, the methodology employs a customized version of the non-max suppression algorithm (NMS) to reduce the number of detections to the most probable ones. For this purpose, the algorithm multiplies objectness probability p_o and class probability vector p_c and takes its maximum, $p_d^* = \max(p_d) = \max(p_o * p_c)$, which it subsequently uses as one decision criterion for reducing detections. This procedure ultimately results in significantly fewer and the most confident detections.

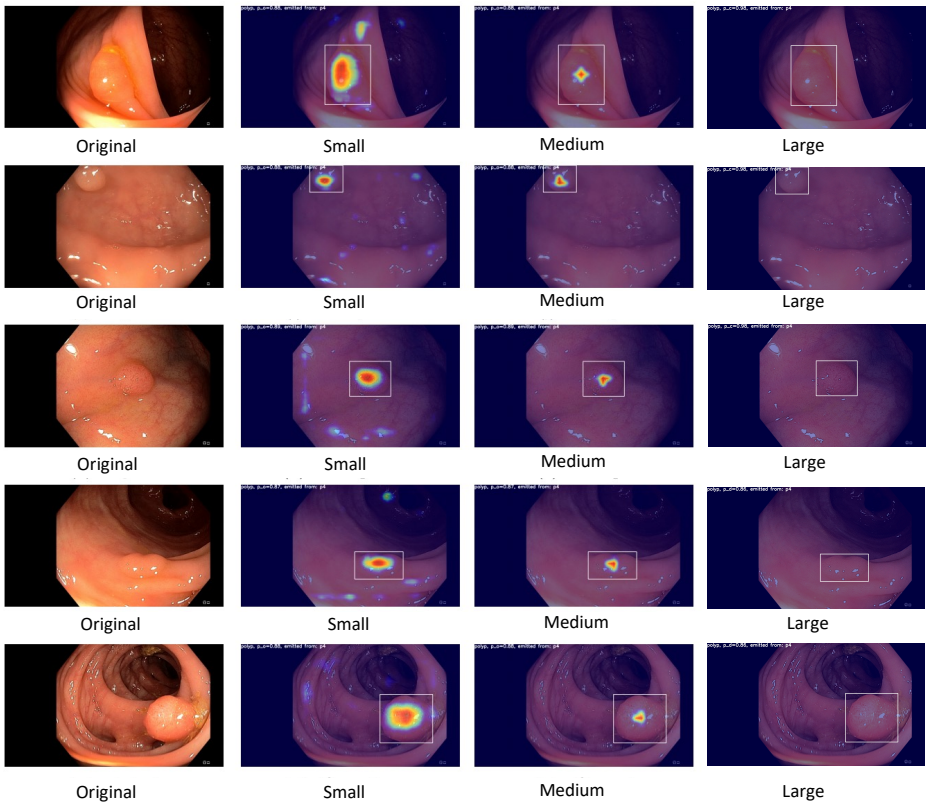


Figure 13 Heatmaps for polyp detection. This figure illustrates the detections of the model using the GRAD-CAM algorithm. Thereby, pixels most relevant for the detection are marked in warm colors like red, and pixels less relevant for the neural network in cold colors like blue. The CNN has three detection outputs for small, medium and large objects.

Furthermore, it associates each with a unique detection probability p_d^*, objectness probability p_o, and class probability $p_c^{(i)}, i = 1 \dots n_c$. The presented methodology carries these values along and inserts them in the Grad-CAM algorithm for y^c.

The next step encompasses the execution of the Grad-CAM algorithm for each of the probabilities mentioned above. Here, the proposed methodology calculates for each probability the gradients $\frac{\partial y^c}{\partial A^k}$ for three feature map activations, namely for p3, p4, and p5. This approach demonstrates the decision-making sequence for detections stemming from p4 and p5.

Afterward, the presented approach transforms the emitted localization maps into heatmaps whose sizes are significantly smaller than the original size of the input image. Therefore, the proposed method upscales them to the original image size by interpolation and then superimposes the heatmaps onto the original image. The resulting image shows highlighted image regions that contain pixels that positively influence the value of y^c. The method also draws for each detection the corresponding bounding box, its score, and the originating scale onto the superimposed image to increase the informational content. The final result is $|\#scores| \times |\#dets| \times |\#scales|$ superimposed images for each input image.

The author of YOLOv5 chose to implement it in the python programming language applying the PyTorch deep learning library. For this reason, this work likewise utilizes this programming language and library to implement the Grad-CAM algorithm for YOLOv5 and necessary extensions. The most worth mentioning feature of the PyTorch deep learning library is the concept of so-called hooks, which enable the extraction of the gradients obtained via backpropagation.

Figure 13 shows in its first column the five original images on which the model should detect the single polyp. The second, third, and fourth columns in figure 13 depict the resulting heatmaps of the approach when assigning p_c to y^c for backpropagation of the respective gradients.

Figure 13 small and 13 medium exhibit the following behaviour: the model increases its focus on the image regions that are crucial for the classification and localization of the polyp present in the image while traversing from output scale small to medium. This observation means that the model increases the pixel intensity crucial for the localization from p3 to p4 as intuitively expected. Moreover, the model works as it should. Furthermore, notice that the highlighted regions in figure 13 encompass the center point of the respective bounding box, which means that the model not only decreases the number of pixels to intensify but also contracts this area to the center point, decreasing the uncertainty of its prediction.

Nevertheless, figure 13 large seem to display the opposite behavior where the detected polyp is not highlighted in the heatmaps. The polyps detected are not large enough to activate the neurons for this particular part of the YOLOv5 architecture.

The observations mentioned above lead to the conclusion that the model work as expected. This fact underlines the necessity of the proposed method to confirm or rebuke the assumption of the analyzed model's proper functioning and expected behavior.

Discussion

In this chapter, we discuss the limitations and clinical use of the system. We especially focus on wrong detections of the polyp detection system and discuss those system failures on our used data sets. Additionally, we debate the clinical application of the system.

Limitations

We initiate the discussion of our limitations with a failure analysis of our model. First, we refer to tables 6 and 10, specifically to videos with significantly worse performance compared to the rest, i.e. videos 8, 12, 15 and 17 of the CVC-VideoClinicDB data set and video 7 of EndoData. Videos in general differ in difficulty even for a human, where some videos only contain obvious polyps in perfect view, while other videos only find them in bad contrast, slanted angles and uncommon shapes. Hence, multiple reasons can be attributed to the worse performance on these videos:

Contrast and lighting are one of the main causes for missing or misidentifying a polyp. Figure 14 shows three frames taken from video 12 of the CVC-VideoClinicDB data set. The image on the left shows a correct detection of a polyp and represents the exception. Most other frames either misidentify the size, as in the middle image,

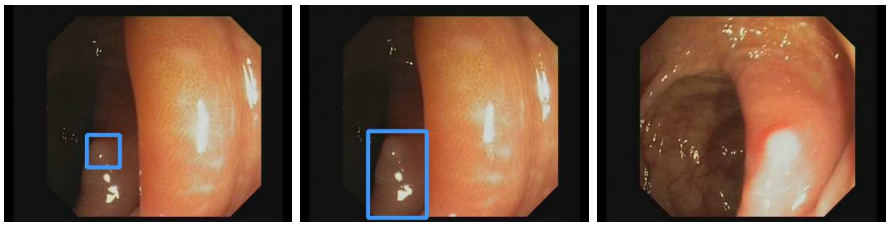


Figure 14 Examples of errors in video 12 of the CVC-VideoClinicDB data set. The left image shows a proper detection, the middle image misidentifies the size of the polyp and in the right image there is no detection due to oversaturation.

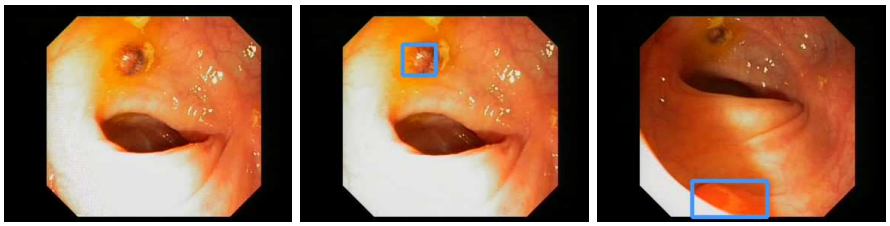


Figure 15 Examples of errors in video 15 of the CVC-VideoClinicDB data set. The left image shows a missed and the middle image a proper detection. On the right image, another polyp in the same frame is detected, while the other one is missed.

or do not detect the polyp at all, as seen on the right. In this case it is most likely an issue of contrast, as the polyp is clearly oversaturated. As this applies to most frames of the video, the F1 score suffers greatly.

Some polyps appear in uncommon shapes, have an atypical surface texture or a rare color and as such are not well represented in the data set. A notable example of such a polyp can be seen in video 15 of the CVC-VideoClinicDB data set with some frames shown in figure 15. Due to its peculiar appearance, this polyp is missed in most frames, especially in those containing another polyp, as seen in the right image. However, the CVC-VideoClinicDB ground truth is partly to blame, since the ground truth masks of this video only cover one polyp at a time even if both are visible in a frame. Nevertheless, rare polyps are a common issue of any supervised model, where the only real improvement is to collect more data of those rare polyps.

Even common polyps can be difficult to detect when obstructed by parts of the colon or due to bad lighting and angles as mentioned above. Figure 16 shows a good example of these issues. An additional problem of polyps such as this one is its small size and color that is very similar to the surrounding colon, leading to a general miss-rate of around 10% of its frames.

Additionally, false positives account for a large amount of errors and drag the evaluation scores down. Often times, areas of the colon look similar to polyps due to lighting and contrast, leading to a false bounding box and dragging down the mAP as well as F1 scores. The number of false positives can be controlled by the user by adjusting the probability threshold for discarding bounding boxes. A large threshold will reduce the number of FPs, but also lead to a larger number of missed polyps and vice versa. For our application, we accept the trade-off of more FPs for

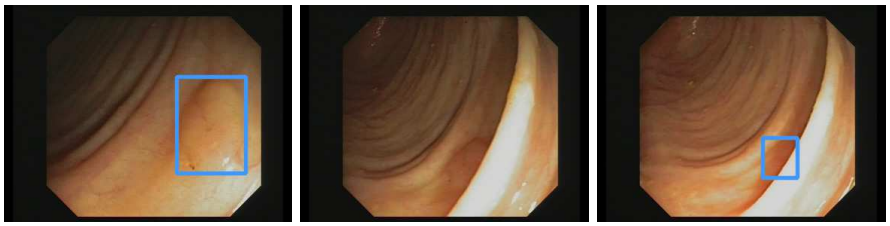


Figure 16 Examples of errors in video 17 of the CVC-VideoClinicDB data set. The left image shows a clean detection of a rather flat polyp. The middle image shows a miss of the same polyp due to being blocked by a colon wall, while the right shows a (short) re-detection.

not missing any polyp in clinical practice. We discuss this point further in the next subsection.

Furthermore, our evaluation shows a significant advantage in using REPP and RT-REPP to reduce the number of FPs. Nevertheless, in a small number of cases, the FPs are increasing when using REPP or RT-REPP e.g in video 9 or video 1 in table 9. This can happen if a false detection is very significant. E.g., the YOLOv5 architecture predicts a box on a bubble and this bubble stays in the frame while not moving. In case the detection score is high REPP and RT-REPP will include FP because it thinks the bubble is solid polyp detection. Nevertheless, REPP and RT-REPP are still reducing small false positives which are seen for several frames. These FPs are still included in the YOLOv5 and Faster-RCNN architecture. Therefore, in special cases, FP can be increased. Nevertheless, we think that these FPs staying longer are less distracting than FPs with a short duration which might mislead the endoscopist and therefore increase withdrawal time of the colonoscopy.

Finally, for clinical application financial cost plays a big role in the usability of our system. We need to ensure a real-time application and as such cannot ensure a server based solution since the delay is simply too large during examination. Hence, each colonoscopy room needs their own system, complete with one or more GPUs. Since the real-time detection needs both fast processing speed and enough VRAM for the video stream, epsecially with RT-REPP, the GPU with the best performance to cost ratio currently is the Nvidia Geforce RTX 3080, with a price of around $800 in December 2021. Depending on the size of the clinic, the cost will easily reach several thousands. However, new GPUs are constantly developed, making current GPUs less expensive.

Clinical Use

A big advantage of our system is that it is already fully implemented as a complete package instead of having several conceptual parts which need to be fitted for the proper system first. As described before, the system fits right between the video stream from an endoscopy camera, processes the input and displays the image on the clinical monitor. Naturally, the direct video stream can still be displayed without our processing on a second monitor. Due to our multi-threaded implementation, the processed image is displayed essentially latency-free, which is a must in the clinical setting. Additionally due to this implementation, in the future slower, more computationally heavy models can be used without having the disadvantage of

higher latency. The system is also applicable to the most commonly used endoscopy processors, expecting a resolution of 1920×1080 pixels. Hence, the system can be set up easily in any common clinical setting.

As mentioned above, false positives are a topic of discussion for evaluation metrics in the context of clinical practice. While, of course, a perfect model would produce only true positives, any trained model will never achieve such performance. In clinical practice, however, it is much worse for the model to produce a false negative rather than a false positive - to a certain degree naturally. Any wrong boxes can be easily checked and confirmed by the doctor to be false, whereas a missed polyp may turn out to be fatal for the patient. As such, while these common metrics essentially weight false positives and negatives the same, clinical practice requires much more weighting on false negatives in order to properly score model performance. As such, while our model does produce false positives, it does not limit the clinical application too much.

Our code is open source and as such, any IT worker should be able to compile and install all necessary components by themselves. However, not every clinic has the necessary resources and experts for this task. While a remote solution can work, as of now, in case of problems and software updates, our dedicated software engineer needs to visit each clinic locally and apply the necessary fixes live since this solution is currently the fastest. We are working on a solution to make updates more dynamic and installable for any clinical environment.

Conclusion

In this study we have implemented and tested a fully assembled real-time polyp detection system that can be used directly in clinical application. For this cause, we have developed and tested an object detection system, the core of our application, which consists of a YOLOv5 object detection CNN and our novel post processing step RT-REPP, a modified REPP for real-time detection. The system was tested on both CVC-VideoClinicDB and our own newly collected and annotated data set (EndoData) and surpassed state-of-the-art detectors with an F1-score of 90.25% while still maintaining real-time speed.

Furthermore, we introduced a new performance metric "first detection time", which measures the time between the first appearance of a polyp and the time of the first detection by the system. We discussed, why the trade-off of a higher number of FPs in return for a better recall is more important for clinical application and hence why this metric is closer to measuring model performance in clinical application.

We have explained and discussed in detail how our full system is assembled and implemented. The direct advantages are flexibility from open source installation and out-of-the-box application set between the endoscopy video stream and the clinic monitor for an almost latency-free bounding box detection display. While there still remain logistic disadvantages, like the need for on-site visits for maintenance, we are working on finding solutions for these issues.

Acknowledgements

We kindly thank the University Hospital of Würzburg, the Interdisziplinäres Zentrum für Klinische Forschung (IZKF) and the Forum Gesundheitsstandort Baden-Württemberg for supporting the research. Furthermore, the authors acknowledge the support by Prof. J.F. Riemann, "Stiftung Lebensblicke" the Foundation for early detection of colon cancer.

Funding
AH and WGZ receive public funding from the state government of Baden-Württemberg, Germany (Funding cluster Forum Gesundheitsstandort Baden-Württemberg) to research and develop artificial intelligence applications for polyp detection in screening colonoscopy. FP receives funding by Interdisziplinäres Zentrum für Klinische Forschung (IZKF) from the University of Würzburg.

Abbreviations
CRC: Colorectal cancer; CNN: Convolutional neural network; CAD: Computer-aided detection; CADx: Computer-aided diagnosis; SSD: Single-shot detector; REPP: Robust and efficient post-processing; RT-REPP: Real-time Robust and efficient post-processing; COCO: Common Objects in Context; JSON: JavaScript Object Notation; YOLO: You Only Look Once; YOLOv5: You Only Look Once (version 5); FDT: Fist Detection Time; AI: Artificial intelligence; GIANA: Gastrointestinal image analysis; WCE: Wireless Capsule Endoscopy; CEM: Context enhancement module; GAN Generative Adversarial Network; FastCat: Fast Colonoscopy Annotation Tool; FPS: Frames per second; GPU: Graphical processing unit; R-CNN: Region based convolutional neural network;

Availability of data and materials
The first data set used for the analysis of this article is available in the GIANA challenge repository (https://endovissub2017-giana.grand-challenge.org/). The second data set (EndoData) used during the analysis is available from the corresponding author on reasonable request.

Ethics approval and consent to participate
The study including retrospective and prospective collection of examination videos and reports was approved by the responsible institutional review board (Ethical committee Landesärztekammer Baden-Württemberg, 21st of January 2021, F-2020-158). All methods were carried out in accordance with relevant guidelines and regulations.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Authors' contributions
AK implemented and coordinated the study, drafted the manuscript, interpreted the data and implemented the software. MB contributed to the creation of the prototype. KM helped with the implementation of the software. AH1, KM, MB contributed to the completion of the manuscript. DF helped with the creation of the data. JT helped with the data preprocessing. BS contributed to the installation of the software. FP, AH2 and WZ provided funding and reviewed the manuscript. All authors read and approved the final manuscript.

Author details
[1]Department of Artificial Intelligence and Knowledge Systems, Sanderring 2, 97070 Würzburg, Germany. [2]Interventional and Experimental Endoscopy (InExEn), Department of Internal Medicine II, University Hospital Würzburg, Oberdürrbacher Straße 6, 97080 Würzburg, Germany. [3]Department of Internal Medicine and Gastroenterology, Katharinenhospital, Kriegsbergstrasse 60, 70174 Stuttgart, Germany.

References
1. Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R.L., Torre, L.A., Jemal, A.: Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians **68**(6), 394–424 (2018). doi:10.3322/caac.21492
2. Mayo Clinic: Colonoscopy. https://www.mayoclinic.org/tests-procedures/colonoscopy/about/pac-20393569 Accessed 2020-10-12
3. Mayo Clinic: Colon Polyps. https://www.mayoclinic.org/diseases-conditions/colon-polyps/symptoms-causes/syc-20352875 Accessed 2021-10-12
4. Rex, D., Cutler, C., Lemmel, G., Rahmani, E., Clark, D., Helper, D., Lehman, G., Mark, D.: Colonoscopic miss rates of adenomas determined by back-to-back colonoscopies. Gastroenterology **112**(1), 24–28 (1997). doi:10.1016/s0016-5085(97)70214-2
5. Heresbach, D., Barrioz, T., Lapalus, M., Coumaros, D., Bauret, P., Potier, P., Sautereau, D., Boustière, C., Grimaud, J., Barthélémy, C., Sée, J., Serraj, I., D'Halluin, P., Branger, B., Ponchon, T.: Miss rate for colorectal neoplastic polyps: a prospective multicenter study of back-to-back video colonoscopies. Endoscopy **40**(04), 284–290 (2008). doi:10.1055/s-2007-995618
6. Leufkens, A., van Oijen, M., Vleggaar, F., Siersema, P.: Factors influencing the miss rate of polyps in a back-to-back colonoscopy study. Endoscopy **44**(05), 470–475 (2012). doi:10.1055/s-0031-1291666
7. van Rijn, J.C., Reitsma, J.B., Stoker, J., Bossuyt, P.M., van Deventer, S.J., Dekker, E.: Polyp miss rate determined by tandem colonoscopy: A systematic review. The American Journal of Gastroenterology **101**(2), 343–350 (2006). doi:10.1111/j.1572-0241.2006.00390.x
8. Kim, N.H., Jung, Y.S., Jeong, W.S., Yang, H.-J., Park, S.-K., Choi, K., Park, D.I.: Miss rate of colorectal neoplastic polyps and risk factors for missed polyps in consecutive colonoscopies. Intestinal research **15**, 411–418 (2017). doi:10.5217/ir.2017.15.3.411
9. Ahn, S.B., Han, D.S., Bae, J.H., Byun, T.J., Kim, J.P., Eun, C.S.: The miss rate for colorectal adenoma determined by quality-adjusted, back-to-back colonoscopies. Gut and liver **6**, 64–70 (2012). doi:10.5009/gnl.2012.6.1.64
10. van Rijn, J.C., Reitsma, J.B., Stoker, J., Bossuyt, P.M., van Deventer, S.J., Dekker, E.: Polyp miss rate determined by tandem colonoscopy: a systematic review. The American journal of gastroenterology **101**, 343–350 (2006). doi:10.1111/j.1572-0241.2006.00390.x

11. Alberto Sabater, A.C.M. Luis Montesano: Robust and efficient post-processing for video object detection. In: International Conference of Intelligent Robots and Systems (IROS) (2020)
12. Krishnan, S.M., Yang, X., Chan, K.L., Kumar, S., Goh, P.M.Y.: Intestinal abnormality detection from endoscopic images. In: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), vol. 2, pp. 895–8982 (1998). doi:10.1109/IEMBS.1998.745583
13. Karkanis, S., Iakovidis, D., Maroulis, D., Karras, D., Tzivras, M.: Computer-aided tumor detection in endoscopic video using color wavelet features. Information Technology in Biomedicine, IEEE Transactions on **7**, 141–152 (2003). doi:10.1109/TITB.2003.813794
14. Hwang, S., Oh, J., Tavanapong, W., Wong, J., de Groen, P.C.: Polyp detection in colonoscopy video using elliptical shape feature. In: 2007 IEEE International Conference on Image Processing, vol. 2, pp. 465–468 (2007). doi:10.1109/ICIP.2007.4379193
15. Bernal, J., Sanchez, J., Vilariño, F.: Towards automatic polyp detection with a polyp appearance model. Pattern Recognition **45**, 3166–3182 (2012). doi:10.1016/j.patcog.2012.03.002,
16. Iakovidis, D.K., Koulaouzidis, A.: Automatic lesion detection in capsule endoscopy based on color saliency: closer to an essential adjunct for reviewing software. Gastrointestinal endoscopy **80**(5), 877–883 (2014)
17. Ratheesh, A., Soman, P., Nair, M.R., Devika, R., Aneesh, R.: Advanced algorithm for polyp detection using depth segmentation in colon endoscopy. In: 2016 International Conference on Communication Systems and Networks (ComNet), pp. 179–183 (2016). IEEE
18. Klare, P., Sander, C., Prinzen, M., Haller, B., Nowack, S., Abdelhafez, M., Poszler, A., Brown, H., Wilhelm, D., Schmid, R.M., *et al.*: Automated polyp detection in the colorectum: a prospective study (with videos). Gastrointestinal endoscopy **89**(3), 576–582 (2019)
19. Yuan, Y., Qin, W., Ibragimov, B., Zhang, G., Han, B., Meng, M.Q.-H., Xing, L.: Densely connected neural network with unbalanced discriminant and category sensitive constraints for polyp recognition. IEEE Transactions on Automation Science and Engineering **17**(2), 574–583 (2020). doi:10.1109/tase.2019.2936645
20. Liu, Y., Tian, Y., Maicas, G., Pu, L.Z.C.T., Singh, R., Verjans, J.W., Carneiro, G.: Photoshopping colonoscopy video frames. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, ??? (2020). doi:10.1109/isbi45749.2020.9098406. https://doi.org/10.1109/isbi45749.2020.9098406
21. Wang, D., Zhang, N., Sun, X., Zhang, P., Zhang, C., Cao, Y., Liu, B.: Afp-net: Realtime anchor-free polyp detection in colonoscopy. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 636–643 (2019). IEEE
22. Liu, M., Jiang, J., Wang, Z.: Colonic polyp detection in endoscopic videos with single shot detection based deep convolutional neural network. IEEE Access **7**, 75058–75066 (2019)
23. Zhang, P., Sun, X., Wang, D., Wang, X., Cao, Y., Liu, B.: An efficient spatial-temporal polyp detection framework for colonoscopy video. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1252–1259 (2019). IEEE
24. Zheng, Y., Zhang, R., Yu, R., Jiang, Y., Mak, T.W., Wong, S.H., Lau, J.Y., Poon, C.C.: Localisation of colorectal polyps by convolutional neural network features learnt from white light and narrow band endoscopic images of multiple databases. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4142–4145 (2018). IEEE
25. Mo, X., Tao, K., Wang, Q., Wang, G.: An efficient approach for polyps detection in endoscopic videos based on faster r-cnn. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3929–3934 (2018). IEEE
26. Zhu, R., Zhang, R., Xue, D.: Lesion detection of endoscopy images based on convolutional neural network features. In: 2015 8th International Congress on Image and Signal Processing (CISP), pp. 372–376 (2015). doi:10.1109/CISP.2015.7407907
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
28. Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., Liang, J.: Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Transactions on Medical Imaging **35**(5), 1299–1312 (2016). doi:10.1109/tmi.2016.2535302
29. Yuan, Z., IzadyYazdanabadi, M., Mokkapati, D., Panvalkar, R., Shin, J.Y., Tajbakhsh, N., Gurudu, S., Liang, J.: Automatic polyp detection in colonoscopy videos. In: Medical Imaging 2017: Image Processing, vol. 10133, p. 101332 (2017). International Society for Optics and Photonics
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (2017). doi:10.1109/tpami.2016.2577031
31. Shin, Y., Qadir, H.A., Aabakken, L., Bergsland, J., Balasingham, I.: Automatic colon polyp detection using region based deep CNN and post learning approaches. IEEE Access **6**, 40950–40962 (2018). doi:10.1109/access.2018.2856402
32. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. CoRR **abs/1602.07261** (2016). 1602.07261
33. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
34. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. ArXiv **abs/1512.02325** (2016)
35. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
36. Zhang, X., Zou, J., He, K., Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 1943–1955 (2016)
37. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818–2826 (2016)
38. Zhang, X., Chen, F., Yu, T., An, J., Huang, Z., Liu, J., Hu, W., Wang, L., Duan, H., Si, J.: Real-time gastric

polyp detection using convolutional neural networks. PLOS ONE **14**(3), 0214133 (2019). doi:10.1371/journal.pone.0214133
39. Zhang, X., Chen, F., Yu, T., An, J., Huang, Z., Liu, J., hu, W., Wang, L., Duan, H., Si, J.: Real-time gastric polyp detection using convolutional neural networks. PloS one **14**, 0214133 (2019). doi:10.1371/journal.pone.0214133
40. Bagheri, M., Mohrekesh, M., Tehrani, M., Najarian, K., Karimi, N., Samavi, S., Reza Soroushmehr, S.M.: Deep neural network based polyp segmentation in colonoscopy images using a combination of color spaces. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6742–6745 (2019). doi:10.1109/EMBC.2019.8856793
41. Sornapudi, S., Meng, F., Yi, S.: Region-based automated localization of colonoscopy and wireless capsule endoscopy polyps. Applied Sciences **9**, 2404 (2019)
42. Yuan, Y., Meng, M.Q.-H.: Deep learning for polyp recognition in wireless capsule endoscopy images. Medical Physics **44**(4), 1379–1389 (2017). doi:10.1002/mp.12147. https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1002/mp.12147
43. Ng, A., *et al.*: Sparse autoencoder. CS294A Lecture notes **72**(2011), 1–19 (2011)
44. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. ArXiv **abs/1701.07875** (2017)
45. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2261–2269 (2017)
46. Zhang, Y., Liu, H., Hu, Q.: Transfuse: Fusing transformers and cnns for medical image segmentation. CoRR **abs/2102.08005** (2021). 2102.08005
47. Liu, X., Guo, X., Liu, Y., Yuan, Y.: Consolidated domain adaptive detection and localization framework for cross-device colonoscopic images. Medical image analysis **71**, 102052 (2021)
48. Qadir, H.A., Balasingham, I., Solhusvik, J., Bergsland, J., Aabakken, L., Shin, Y.: Improving automatic polyp detection using CNN by exploiting temporal dependency in colonoscopy video. IEEE Journal of Biomedical and Health Informatics **24**(1), 180–193 (2020). doi:10.1109/jbhi.2019.2907434
49. Misawa, M., Kudo, S., Mori, Y., Cho, T., Kataoka, S., Maeda, Y., Ogawa, Y., Takeda, K., Nakamura, H., Ichimasa, K., *et al.*: Tu1990 artificial intelligence-assisted polyp detection system for colonoscopy, based on the largest available collection of clinical video data for machine learning. Gastrointestinal Endoscopy **89**(6), 646–647 (2019)
50. Itoh, H., Roth, H., Oda, M., Misawa, M., Mori, Y., Kudo, S.-E., Mori, K.: Stable polyp-scene classification via subsampling and residual learning from an imbalanced large dataset. Healthcare Technology Letters **6**(6), 237–242 (2019). doi:10.1049/htl.2019.0079
51. Qadir, H.A., Balasingham, I., Solhusvik, J., Bergsland, J., Aabakken, L., Shin, Y.: Improving automatic polyp detection using cnn by exploiting temporal dependency in colonoscopy video. IEEE Journal of Biomedical and Health Informatics **24**(1), 180–193 (2020). doi:10.1109/JBHI.2019.2907434
52. Bernal, J., Sánchez, J., Vilarino, F.: Towards automatic polyp detection with a polyp appearance model. Pattern Recognition **45**(9), 3166–3182 (2012)
53. Silva, J., Histace, A., Romain, O., Dray, X., Granado, B.: 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 (2014)
54. Silva, J., Histace, A., Romain, O., Dray, X., Granado, B.: Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. International journal of computer assisted radiology and surgery **9**, 283–293 (2014). doi:10.1007/s11548-013-0926-3
55. Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., Vilariño, F.: WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. Computerized Medical Imaging and Graphics **43**, 99–111 (2015). doi:10.1016/j.compmedimag.2015.02.007
56. Angermann, Q., Bernal, J., Sánchez-Montes, C., Hammami, M., Fernández-Esparrach, G., Dray, X., Romain, O., Sánchez, F.J., Histace, A.: Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis. In: Cardoso, M.J., Arbel, T., Luo, X., Wesarg, S., Reichl, T., González Ballester, M.Á., McLeod, J., Drechsler, K., Peters, T., Erdt, M., Mori, K., Linguraru, M.G., Uhl, A., Oyarzun Laura, C., Shekhar, R. (eds.) Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures, pp. 29–41. Springer, Cham (2017)
57. Vázquez, D., Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., López, A.M., Romero, A., Drozdzal, M., Courville, A.: A benchmark for endoluminal scene segmentation of colonoscopy images. Journal of Healthcare Engineering **2017**, 1–9 (2017). doi:10.1155/2017/4037190
58. Vázquez, D., Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., López, A.M., Romero, A., Drozdzal, M., Courville, A.: A benchmark for endoluminal scene segmentation of colonoscopy images. Journal of healthcare engineering **2017**, 4037190 (2017). doi:10.1155/2017/4037190
59. Bernal, J., Sanchez, J., Vilariño, F.: Towards automatic polyp detection with a polyp appearance model. Pattern Recognition **45**, 3166–3182 (2012). doi:10.1016/j.patcog.2012.03.002,
60. Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., Vilariño, F.: Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society **43**, 99–111 (2015). doi:10.1016/j.compmedimag.2015.02.007
61. Fernández-Esparrach, G., Bernal, J., López-Cerón, M., Córdova, H., Sánchez-Montes, C., Rodríguez de Miguel, C., Sánchez, F.J.: Exploring the clinical potential of an automatic colonic polyp detection method based on the creation of energy maps. Endoscopy **48**, 837–842 (2016). doi:10.1055/s-0042-108434
62. Jha, D., Smedsrud, P.H., Riegler, M.A., Halvorsen, P., de Lange, T., Johansen, D., Johansen, H.D.: Kvasir-seg: A segmented polyp dataset. In: International Conference on Multimedia Modeling, pp. 451–462 (2020). Springer
63. Misawa, M., Kudo, S.-e., Mori, Y., Hotta, K., Ohtsuka, K., Matsuda, T., Saito, S., Kudo, T., Baba, T., Ishida, F., *et al.*: 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 (2021)
64. Ali, S., Braden, B., Lamarque, D., Realdon, S., Bailey, A., Cannizzaro, R., Ghatwary, N., Rittscher, J., Daul, C., East, J.: Endoscopy Disease Detection and Segmentation (EDD2020). IEEE Dataport (2020). doi:10.21227/f8xg-wb80. http://dx.doi.org/10.21227/f8xg-wb80
65. Krenzer, A., Makowski, K., Hekalo, A., Fitting, D., Troya, J., Zoller, W.G., Hann, A., Puppe, F.: Fast machine learning annotation in the medical domain: A semi-automated video annotation tool for gastroenterologists (2021)
66. Lambert, R.f.: Endoscopic classification review group. update on the paris classification of superficial neoplastic lesions in the digestive tract. Endoscopy **37**(6), 570–578 (2005)
67. Kang, J., Gwak, J.: Ensemble of instance segmentation models for polyp segmentation in colonoscopy images. IEEE Access **7**, 26440–26447 (2019). doi:10.1109/access.2019.2900672
68. Jha, D., Smedsrud, P.H., Riegler, M.A., Johansen, D., De Lange, T., Halvorsen, P., Johansen, H.D.: Resunet++: An advanced architecture for medical image segmentation. In: 2019 IEEE International Symposium on Multimedia (ISM), pp. 225–2255 (2019). IEEE
69. Guo, Y.B., Matuszewski, B.: Giana polyp segmentation with fully convolutional dilation neural networks. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 632–641 (2019). SCITEPRESS-Science and Technology Publications
70. de Almeida Thomaz, V., Sierra-Franco, C.A., Raposo, A.B.: Training data enhancements for robust polyp segmentation in colonoscopy images. In: 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), pp. 192–197 (2019). IEEE
71. Qadir, H.A., Solhusvik, J., Bergsland, J., Aabakken, L., Balasingham, I.: A framework with a fully convolutional neural network for semi-automatic colon polyp annotation. IEEE Access **7**, 169537–169547 (2019). doi:10.1109/access.2019.2954675
72. Ali, S., Zhou, F., Daul, C., Braden, B., Bailey, A., Realdon, S., East, J., Wagnières, G., Loschenov, V., Grisan, E., Blondel, W., Rittscher, J.: Endoscopy artifact detection (EAD 2019) challenge dataset. CoRR **abs/1905.03209** (2019). 1905.03209
73. Soberanis-Mukul, R.D., Kayser, M., Zvereva, A.A., Klare, P., Navab, N., Albarqouni, S.: A learning without forgetting approach to incorporate artifact knowledge in polyp localization tasks. ArXiv **abs/2002.02883** (2020)
74. Xu, R., Lin, H., Lu, K., Cao, L., Liu, Y.: A forest fire detection system based on ensemble learning. Forests **12**(2), 217 (2021)
75. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014). Springer
76. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
77. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
78. Mongan, J., Moy, L., Kahn Jr, C.E.: Checklist for artificial intelligence in medical imaging (CLAIM): a guide for authors and reviewers. Radiological Society of North America (2020)