An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification

VAJIRA THAMBAWITA, SimulaMet and Oslo Metropolitan University
DEBESH JHA, SimulaMet and UiT—The Arctic University of Norway
HUGO LEWI HAMMER, Oslo Metropolitan University and SimulaMet
HÅVARD D. JOHANSEN and DAG JOHANSEN, UiT—The Arctic University of Norway
PÅL HALVORSEN, SimulaMet and Oslo Metropolitan University
MICHAEL A. RIEGLER, SimulaMet

Precise and efficient automated identification of gastrointestinal (GI) tract diseases can help doctors treat more patients and improve the rate of disease detection and identification. Currently, automatic analysis of diseases in the GI tract is a hot topic in both computer science and medical-related journals. Nevertheless, the evaluation of such an automatic analysis is often incomplete or simply wrong. Algorithms are often only tested on small and biased datasets, and cross-dataset evaluations are rarely performed. A clear understanding of evaluation metrics and machine learning models with cross datasets is crucial to bring research in the field to a new quality level. Toward this goal, we present comprehensive evaluations of five distinct machine learning models using global features and deep neural networks that can classify 16 different key types of GI tract conditions, including pathological findings, anatomical landmarks, polyp removal conditions, and normal findings from images captured by common GI tract examination instruments. In our evaluation, we introduce performance hexagons using six performance metrics, such as recall, precision, specificity, accuracy, F1-score, and the Matthews correlation coefficient to demonstrate how to determine the real capabilities of models rather than evaluating them shallowly. Furthermore, we perform cross-dataset evaluations using different datasets for training and testing. With these cross-dataset evaluations, we demonstrate the challenge of actually building a generalizable model that could be used across different hospitals. Our experiments clearly show that more sophisticated performance metrics and evaluation methods need to be applied to get reliable models rather than depending on evaluations of the splits of the same dataset—that is, the performance metrics should always be interpreted together rather than relying on a single metric.

CCS Concepts: • **Computing methodologies** → **Cross-validation**; **Supervised learning by classification**; **Machine learning approaches**; • **Applied computing** → **Life and medical sciences**;

17

This work was funded in part by the Research Council of Norway under project number 263248 (Privaton).
Authors' addresses: V. Thambawita, SimulaMet, Oslo, Norway, and Oslo Metropolitan University, Oslo, Norway; email: vajira@simula.no; D. Jha, SimulaMet, Oslo, Norway, and UiT—The Arctic University of Norway, Tromsø, Norway; email: debesh@simula.no; H. L. Hammer, Oslo Metropolitan University, Norway, and SimulaMet, Oslo, Norway; email: hugoh@oslomet.no; H. D. Johansen and D. Johansen, UiT—The Arctic University of Norway, Tromsø, Norway; emails: {havard.johansen, dag.johansen}@uit.no; P. Halvorsen, SimulaMet, Oslo, Norway, and Oslo Metropolitan University, Oslo, Norway; email: paalh@simula.no; M. A. Riegler, SimulaMet, Oslo, Norway; email: michael@simula.no.

2637-8051/2020/06-ART17 $15.00
https://doi.org/10.1145/3386295

Additional Key Words and Phrases: Medical, computer-aided diagnosis, global features, deep learning, multi-class classification, gastrointestinal tract diseases, polyp classification, Kvasir, Nerthus, CVC-356, CVC-612, CVC-12K, cross-dataset evaluations

ACM Reference format:
Vajira Thambawita, Debesh Jha, Hugo Lewi Hammer, Håvard D. Johansen, Dag Johansen, Pål Halvorsen, and Michael A. Riegler. 2020. An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification. *ACM Trans. Comput. Healthcare* 1, 3, Article 17 (June 2020), 29 pages.
https://doi.org/10.1145/3386295

1 INTRODUCTION

Cancer is one of the leading causes of death worldwide and a significant barrier to life expectancy [12]. In particular, the gastrointestinal (GI) tract can be affected by a variety of diseases and abnormalities [52]. Using data from the Global Cancer Observatory,[1] Bray et al. [12] estimated that, for 2018, there would be around 5 million new luminal GI cancer incidences and about 3.6 million deaths due to GI cancer.[2] The most frequently diagnosed GI cancers in 2018 for new cases were colorectal cancer (CRC) (6.1%), stomach cancer (5.7%), liver cancer (4.7%), rectum cancer (3.9%), and esophageal cancer (3.2%) out of 36 types of cancers [12].

Gastroscopy and colonoscopy are the most successful medical procedures for GI endoscopy examinations. Among both, colonoscopy has been proven to be an effective preventative method by improving declination in the occurrence of Colorectal Cancer (CRC) by 30% [41]. During a colonoscopic procedure, an endoscopist inserts a colonoscope carefully through the anus to examine the rectum and colon. A tiny wide-angle video camera mounted at the end of the colonoscope captures a live video signal of the internal mucosa of the patient's colon. The endoscopist uses the video signal for real-time diagnosis of the patient, where one of the primary goals is to identify and remove abnormalities such as polyps [77].

The current EU guidelines [74] recommend GI tract screening for all people older than 50 years. Such regular screenings can be of great significance for early detection and prevention of cancer inside the GI tract, but they are challenging due to many factors. Moreover, a colonoscopy examination is entirely an operator-dependent screening procedure [63]. The detection rate of GI tract lesions mostly relies on the clinical experience of the gastroenterologist. The shortage of experienced gastroenterologists, and the clinicians' tiredness and lack of concentration during the colonoscopic examination, can lead to missing polyps that otherwise would be detected [68]. The estimated miss rate for the subject undergoing a colonoscopy examination is 25% [39].

Although considerable work has been done to develop and improve systems for automatic polyp detection, the performance of existing solutions is still behind that of an expert endoscopist [7, 16, 44, 75, 76]. Most of the published works in the field use non-public datasets or develop models from too-small training, validation, and test sets [7, 75, 76]. The performance metrics used to measure the performance of methods are also not sufficient (e.g., see the first part of Table 1). Thus, it is difficult for researchers to compare and reproduce some of the present related works. Moreover, the state-of-the-art research in this field does not present the generalizability of their solutions using cross-dataset evaluations. As a result, it creates a distrust for applying these machine learning (ML) solutions in practice.

An automatic and efficient computer-aided diagnosis (CAD) system in a clinic could assist medical experts during the endoscopic and colonoscopy procedure to improve the detection rate by finding unrecognized lesions and act as a second observer by providing better insights to the gastroenterologist concerning the presence and types of lesions. With this inspiration, we conducted five experiments to classify 16 classes of GI tract conditions

[1] https://gco.iarc.fr.
[2] We have considered the statistic of esophagus, stomach, colon, rectum, anus, gallbladder, and pancreas.

Table 1. Overview of the Related Work

Reference	Year	REC	PREC	SPEC	ACC	MCC	F1	Rk	FPS
Hwang et al. [27]	2007	0.9600	0.8300	—	—	—	—	—	15
Li & Meng [40]	2012	0.8860	—	0.9620	0.9240	—	—	—	—
Zhou et al. [83]	2014	0.7500	—	0.9592	0.9077	—	—	—	—
Wang et al. [76]	2014	0.8140	—	—	—	—	—	—	0.14
Mamonov et al. [43]	2014	0.4700	—	0.9000	—	—	—	—	—
Wang et al. [77]	2015	0.9770	—	—	0.9570	—	—	—	10
Riegler et al. [57]	2016	0.9850	0.9388	0.7250	0.8770	—	—	—	∼300
Shin & Balasingham [63]	2017	0.9082	0.9271	0.9176	0.9126	—	—	—	—
Riegler et al. [58]	2017	0.9850	0.9390	0.7250	0.8770	—	—	—	∼75
Yu et al. [78]	2017	0.5005	0.4917	—	0.9471	—	0.4830	0.5357	—
Pogorelov et al. [54]	2017	0.8260	0.8290	0.9750	0.9570	—	0.8260	0.8020	46
Agrawal et al. [1]	2017	—	—	—	0.9610	0.8260	0.8470	—	—
Naqvi et al. [45]	2017	—	0.7665	0.9660	0.9420	0.7360	0.7670	—	—
Petscharnig et al. [48]	2017	0.7550	0.7550	0.9650	0.9390	0.7200	0.7550	0.7240	—
Pogorelov et al. [52]	2017	0.9060	0.9060	0.9810	0.9690	—	—	—	30
Yuan et al. [79]	2018	0.8180	0.7232	—	—	—	0.7431	—	—
Wang et al. [75]	2018	0.9438	—	0.9592		—	—	—	—
Mori & Kudo [44]	2018	>0.9000	—	>0.9000	—	—	—	—	—
MediaEval 2018 Medico Task [53] (The following experiments were done using the 2018 Medico dataset.)									
Hoang et al. [25]	2018	0.9281	**0.9426**	**0.9963**	**0.9932**	**0.9312**	**0.9342**	**0.9398**	23
Hicks et al. [24]	2018	0.9218	0.9378	0.9959	0.9924	0.9228	0.9236	0.9325	624
Borgli et al. [10]	2018	0.8572	0.8708	0.9956	0.9918	0.8555	0.8555	0.9280	—
Kirkerød et al. [36]	2018	0.8433	0.8514	0.9944	0.9896	0.8366	0.8367	0.9082	—
Dias & Dias [18]	2018	0.8205	0.8414	0.9938	0.9885	0.8146	0.8114	0.8983	8.61
Taschwer et al. [70]	2018	0.8673	0.8826	0.9933	0.9876	0.8641	0.8662	0.8897	—
Ostroukhova et al. [46]	2018	0.8236	0.8281	0.9911	0.9835	0.8115	0.8145	0.8539	1E-100
Khan & Tahir [33]	2018	0.6203	0.7173	0.9767	0.957	0.6025	0.5868	0.6302	43329
Steiner et al. [64]	2018	0.4219	0.5146	0.9717	0.9469	0.3901	0.3913	0.5368	—
Ko et al. [37]	2018	0.5005	0.4916	0.9715	0.9471	0.4608	0.4829	0.5357	0.5357
Thambawita et al. (Ours) [71]	2018	**0.9361**	0.9319	**0.9963**	**0.9932**	0.9283	0.9297	0.9397	—

REC, recall (sensitivity); ACC, accuracy; MCC, Matthews correlation coefficient; F1, F1-score; Rk, Rk correlation coefficient; FPS, frames per second.

The results of the Medico Task may slightly vary compared to the proceeding note papers because of different ways of calculating the multi-class performance metrics by the organizers. The highest score for the MediaEval 2018 Medico Task is marked in bold.

for the Medico Multimedia Task at MediaEval 2018 [53]. One example for each of the 16 classes is depicted in Figure 1.

In this work, we focus on identifying the limitations of generalizing ML models across different datasets and how to interpret the evaluation metrics in that context. For this, we are using global feature (GF)-based and deep learning (DL)-based methods that performed well at the 2018 Medico Task [53], where one specific dataset was used. In addition, here we explore the different performance metrics of both methods (GF and deep learning (DL)) to identify the limitations of each. We show that combined complex deep neural network (DNN) models outperform other methods. Finally, we explore how multi-class models perform on polyp and non-polyp detection with and without retraining the model for the two specific classes. The effects of retraining for classifying the sub-categories of the same dataset and using them in other datasets are analyzed in detail to identify

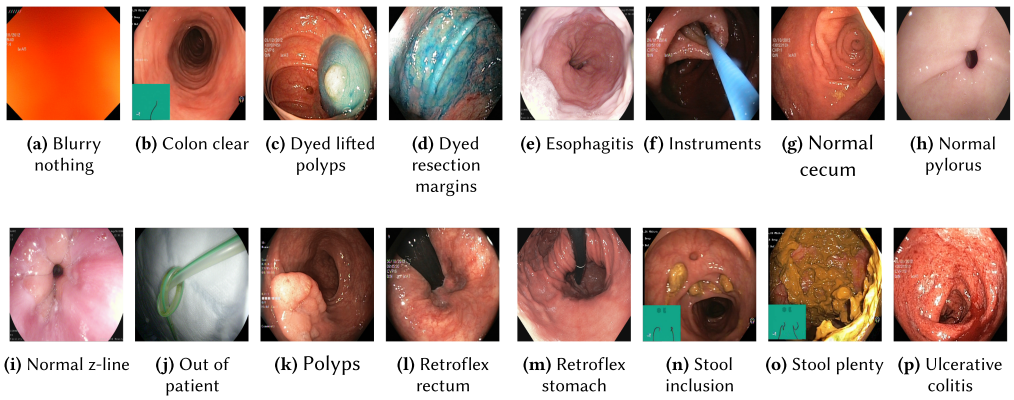


Fig. 1. Sample images of GI findings. Each image represents one of the 16 classes from the dataset used for the Medico 2018 Challenge [50, 51].

the cross-dataset generalization capabilities of our models. We emphasize that a large number of performance measures do not show the real performance of ML models. We also highlight the necessity of having cross-dataset evaluations to determine the real capabilities of ML models before using them in clinical settings.

To study cross-dataset bias and metrics interpretation, our contributions are as follows:

(1) We present five ML classification models to classify multi-class findings (anatomical landmark, pathological findings, polyp removal conditions, and normal findings) of the GI tract. Using a limited imbalanced dataset, we experiment with approaches ranging from Global Feature (GF) approaches to simple Deep Neural Network (DNN) and complex DNN approaches with transfer learning. Moreover, we present a detailed evaluation using six performance metrics to show the real classification performance of ML models. In addition, we analyze and present detailed evaluation results of using multi-class classification ML models for classifying binary classes (sub-classes of the multi-class categories) with and without retraining to evaluate the generalizability of our models. We emphasize the difficulties of using well-performing ML methods in cross-datasets as a result of the reluctance of ML models to cross-dataset generalization. We present this negative impact with the aid of another evaluation using the receiver operating characteristic (ROC) curve and the precision-recall (PR) curve of the best model. We also demonstrate when a Receiver Operating Characteristic (ROC) curve is good to use and when it is better to use a PR curve.
(2) With the preceding point, we emphasize the requirement of detailed cross-dataset evaluations to identify generalizability of ML models before using them as universal models in live applications. Because good performance measures with a single dataset do not necessarily imply good real-world performance, we argue that researchers should present cross-dataset evaluations for building a generalizable model rather than presenting performance values for the test datasets, which is separated from the same training data source.

Moreover, with respect to the 2018 Medico Task [53], our best DNN method achieved the highest recall, specificity, and accuracy for multi-class classification of the GI tract findings. We achieved a Matthews correlation coefficient (MCC) (0.0029 less) and an Rk correlation coefficient[3] (0.0001 less) nearly equal to the winning team. With this achievement, we demonstrate all of the steps, from designing to training and testing, for reaching such performance using this model and its expandability using different pre-trained networks.

[3]The Rk correlation coefficient and the MCC were the most important considered metrics for winning the 2018 Medico Task.

In Section 2, we present related work and the performance of relevant existing solutions. Section 3 discusses the methodology used for our GF-based approaches and the theoretical foundation for our work. The DNN-based approaches are similarly described in Section 4. Our experimental results are presented and analyzed in Section 5, followed by a discussion in Section 6 on how our results can be helpful to other researchers. In Section 7, we conclude our findings.

2 RELATED WORK

Many methods and algorithms have been proposed for GI tract disease detection/classification using videos and images from colonoscopy and gastroscopy as input. The problem of polyp detection has by far received the most attention by researchers. Images and videos of polyps and other abnormalities inside the GI tract are usually collected using a specific-purpose camera and imaging system, like ScopeGuide from Olympus. The information gathered from these types of devices may be of great significance for later examination and must be handled with great care. Polyps generally have characteristics different from the normal surrounding healthy tissue and are often easy for clinicians to detect. There are several good datasets available for training and testing on polyps (the details about the available polyp dataset can be found in other works [16, 30]), and binary classification methods are relatively straightforward to implement.

The other active research efforts include developing an automatic and real-time detection system for GI bleeding, ulcerative lesion, blood-based abnormality, tumor, and angiectasia, and for multi-class data of the GI tract that comprise anatomical landmarks (e.g., z-line, pylorus, and cecum), pathological findings (e.g., esophagitis and ulcerative colitis), and normality and regular findings (e.g., normal colon mucosa and stool). Suitable datasets for research in these areas are less developed and lack adequate content. Similarly, presented performance measures in these areas are not adequate because of not presenting enough performance metrics or not presenting cross-dataset evaluations.

Table 1 presents an overview of important works related to GI disease detection/classification and the 2018 Medico Task [53] using Computer-Aided Diagnosis (CAD), from automatic polyp detection to multi-class disease detection and classification systems. The dataset used for the experiments in the first half of the Table 1 is different. Therefore, the results cannot be directly compared; however, the results in the lower half of the table can be compared, as the algorithms are tested on the same dataset.

Most of the research in the medical field only focuses on designing an automated disease detection system for detecting or classifying specific disease or abnormality, such as polyp detection or ulcer detection. Because patients may suffer from more than one type of disease at the time, a working multi-class disease detection system will help treatment. The performance of existing multi-abnormality detection systems is, however, not satisfactory and cannot assist doctors in CAD in real time while undergoing colonoscopies. Furthermore, these research works have not evaluated all performance metrics at once to analyze the real behavior of their classification models. Yet none of the preceding methods have performed cross-dataset evaluations to prove the capabilities for using the ML models in real CAD systems.

For handcrafted (HC) feature-based methods, image descriptors like global or local image features (e.g., color, texture, and edges) are extracted, and later on, various ML classifiers (e.g., logistic model tree (LMT) [71], random forest classifier [43], or support vector machine (SVM) [76]) are employed to perform analysis using these features. HC descriptors (manually designed features) are useful for the gastroenterologist while identifying specific abnormality regions inside the GI tract. For instance, as blood has a particular range of chromaticity, we can specify a specific chromaticity range where features of bleeding abnormality seem to be concentrated [31]. Riegler et al. [58] achieved an F1-score of 0.909 with a GF-based approach and an F1-score of 0.875 with a DL-based approach with a multi-class GI tract dataset. With the ASU-Mayo polyp dataset, the GF-based approach achieves an F1-score of 0.961, whereas the DL-based approach could obtain 0.936. They further suggested that the combination of both approaches may lead to improved performance. In addition, previous work by Riegler

et al. [56] reveals that although only detecting whether a frame contains an irregularity or not, GFs can beat local features—for instance, they can at least reach the same results with regard to detection/classification and perform better than local features with regard to processing speed. In all of these works, researchers presented performance metrics using a test dataset selected from the same dataset used for the training data. Therefore, these results do not reflect the actual practical performance of the proposed methods.

A few past studies used information such as the color and texture of polyps to sketch HC descriptors [2, 3, 13, 28, 29, 32, 68]. The other category of methods for automated polyp detection used shape, intensity, edge, and spatio-temporal information. For instance, Hwang et al. [27] appropriated elliptical shape features to detect the occurrence of polyps in colonoscopy videos. Bernal et al. [7] proposed a polyp detection technique by utilizing a polyp region descriptor, which is dependent on the depth of the valley image and introduced a region growing method to detect polyps in colonoscopy images. Bernal et al. [8] additionally used valley information and enhanced their approach by improving the polyp localization results to almost 30%. Bernal et al. [6] also performed additional evaluations using valley information and demonstrated better performance, especially for smaller polyps and decreased the polyp miss rate. Park et al. [47] utilized spatio-temporal features for automatic polyp detection. The recently completed related work that uses the cross-sectional profile to detect protruding polyps automatically is the polyp detection system Polyp-Alert [77], which can provide near real-time feedback during colonoscopies. However, the system is limited to polyp detection and is slow for live examinations. Tajbakhsh et al. [68] proposed a method for automatic polyp detection from colonoscopy videos that uses context information to remove non-polyp and shape information to localize polyp reliably. Riegler et al. [58] utilized various GFs and achieved high precision and recall above 90%. Yuan et al. [79] employed a bottom-up and top-down saliency approach for automated polyp detection. Although these research works discuss improving the performance of ML models, they have not evaluated the performance of the ML models with cross-datasets.

As convolutional neural network (CNN) architectures have achieved exceptional gains in medical image and video analysis tasks, more recent work on polyp detection is mainly based on Convolutional Neural Networks (CNNs). Tajbakhsh et al. [67] proposed a 2D-CNN method for polyp detection by learning discriminative spatial and temporal features. Yu et al. [78] proposed a 3D fully convolutional network to deal with the challenges related to automatic polyp detection for colonoscopy videos. Zhang et al. [81] suggested an enhanced single-shot multi-box detector (SSD) called *SSD-GPNet* for detecting gastric polyps, which have the potential for achieving real-time detection up to 50 FPS using Nvidia Titan V. Furthermore, they use GPDNet [82] to classify three classes of pre-cancerous gastric disease.

Researchers have also compared HC and DL methods. For instance, Pogorelov et al. [52] and Riegler et al. [58] compared several (HC- and DL-based) localization methods. Pogorelov et al. [49] evaluated their approach utilizing HC and DL methods on different available datasets for real-time polyp detection. Their best model with a generative adversarial network (GAN) obtained detection specificity of 94% and accuracy of 90.9%. The preceding research works presented good performance for predicting polyps, whereas Pogorelov et al. [49] presented evaluation results of the models with cross-datasets. However, having overlapped data sources in the cross-datasets, the shown results do not reveal the real performance in cross-dataset evaluations.

The pre-trained models, along with transfer learning mechanisms, are also becoming popular because of their capability to outperform state-of-the-art algorithms even with less training data, where the limited size of the medical dataset for experiments has always been a problem to yield better results. For the detection and localization of the polyps [9, 69], the pre-trained models with a CNN mechanism also achieve promising results. A comparison of DL with GFs for GI tract disease detection has also been presented. Pogorelov et al. [54] presented 17 different methods for multi-class classification of GI tract data with the limited number of the training dataset. They used both GFs and DL approaches in their work. They achieved the best result with modified ResNet50 features using the LMT classifier. They reached an Rk value of 80.2% and an F1-score of 82.6% with 2,000 training and 2,000 test datasets.

Comparing with the polyp detection approaches, the research on multi-class disease detection/classification on a complete GI tract system is minimal. However, for multi-class disease detection/classification (including polyp detection) inside the GI tract, we note a few contributions made in this area. For example, the authors of numerous works [1, 10, 18, 24, 25, 33, 36, 37, 46, 48, 64, 70] presented their approach in classifying disease inside the GI tract utilizing the Kvasir dataset and the MediaEval Medico 2018 dataset. The latter is a combination of the Nerthus [50] and Kvasir [51] datasets.

Hicks et al. [24] show how fine tuning a CNN model using transfer learning with data from different source domains affects classification performance. In their case, extending the generic ImageNet dataset with medical images from the LapGyn4 and Cataract-101 dataset, they obtained a high Matthews Correlation Coefficient (MCC) score of 0.9228. For the 2018 Medico Task, we proposed solutions based on GFs and DL-based methods for multi-class classification of GI tract findings [71]. Our best model was a combination of two pre-trained networks, ResNet-152 and Densenet-161, along with a multi-layer perceptron (MLP). Here, we obtained an MCC of 94.21%, an F1-score of 94.58%, and an accuracy of 99.32%. This was one of the best results in the MediEval 2018 Medico Task Challenge. We discuss the model introduced by Thambawita et al. [71] in detail in this article and reproduce similar results. Based on those models, we provide and discuss the requirement of detailed evaluations using multiple performance metrics and cross-dataset evaluations.

Recent related works show promising results in terms of evaluation metrics, such as both sensitivity and specificity despite various challenges (e.g., difficulties arise due to a dataset obtained from different modalities). The limitation with most of the recent approaches is that they target only specific problems, like bleeding detection or polyp detection. Current systems are either (i) too narrow for a flexible, multi-disease detection/classification system; (ii) tested only on a limited datasets, too small to show whether the systems would work well in hospitals, (iii) provide low processing performance for a real-time system or ignore the system performance entirely; (iv) problematic with regard to overfitting of the specific dataset and lead to unreliable results; or (v) tested using datasets that are not publicly available, making it difficult to compare the approaches with others.

In some cases, GF-based approaches produce better results. For some methods, DL performs better. The CNN approaches and pre-trained network with transfer learning mechanism approaches have the best results in most of the cases. Reusing already existing DL architectures and pre-trained models leads to excellent results in, for example, the ImageNet classification tasks. For example, the HC feature-based approach works well for true negative (TN) detection/classification tasks.

To reduce the damage of the dataset bias problem, Khosla et al. [34] directed their experiments for both classification tasks and detection problems. They used different datasets from different domains in the training stage to generalize the features extracted from their ML model. However, SVM was used as the main algorithm, and the DNN dataset bias problem was not addressed.

With the goal of making researchers aware of the dataset bias problems, Torralba and Efros [72] did informative research using basic datasets and basic ML models with the classification and detection task of computer vision. Initially, they trained a simple linear Support Vector Machine (SVM) to make a simple classifier to name a given dataset from 12 different datasets, which have nearly the same categories. They were inspired by the research done by Dollár et al. [20] to detect pedestrians. The result of the experiment for dataset classification shows a clear diagonal in the confusion matrix (CM). This implies that there are clear dataset bias features, and that these datasets have the same categories. Therefore, researchers want to apply cross-dataset generalization for avoiding dataset bias behavior of ML models. Moreover, they discussed selection bias, capture bias, category or label bias, and negative bias as the main factors for the dataset bias. This directs our research to do additional experiments to identify the significant factors of the cross-domain data generalization in the medical domain, which is more critical than the general image classification.

The classification of GI diseases is more complicated than a simple real-world object classification task where one detects faces or recognizes characters. Typical GI tract datasets are heavily imbalanced—for example, the 2018 Medico Task dataset consists of 16 classes of anatomical landmarks, pathological findings, polyp removal

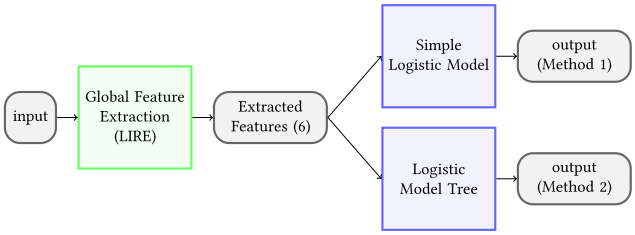


Fig. 2. Block diagram of the proposed method 1 and method 2. The pipeline starts with the input of images. GFs are extracted using the LIRE framework. These features are then used for two different classification algorithms (the SL model for method 1 and LMT for method 2).

cases, and normal and regular findings, where the polyp class has a maximum of 613 images, and the instrument class has a minimum of only 4 images. Additionally, medical datasets are captured using different endoscopic instruments, and some of the images can be noisy, blurry, over- or under-exposed, and interleaved, and can have superfluous information within the image, contain borders, and be affected by specular reflections caused by the instrument light source. Some of the images may have bleeding, whereas other images can be partially covered by stool or mucus. Moreover, the organs from mouth to anus can have multiple lesions showing different diseases, abnormalities, and internal injuries. Thus, the preceding situation leads to the necessity of distinguishing between various classes of GI tract findings. In this scenario, not only high precision and recall but also high accuracy and MCC become essential for developing an automated generalizable multi-class classification system. This implies the real requirement of measuring and analyzing all performance metrics at once. Furthermore, to prove the generalizability of models, cross-dataset evaluations are required.

3 GF-BASED APPROACHES

GFs or descriptors are features computed over the whole image or covering a regular sub-section of an image. GFs represent the overall properties of an image and are often used in image retrieval, image compression, image classification, object detection, and image collection search and distance computing [54]. Examples of GFs are shape matrix, histogram-oriented gradients (HOGs), Co-HOG, and invariant moments (Hu, Zernike). The LIRE [42] framework can be used to extract HC GFs such as texture, color distribution, and the histogram of brightness. The most commonly used GFs include joint composite descriptor (JCD), Tamura, color layout (CL), edge histogram (EH), autocolor correlogram, pyramid histogram of oriented gradients (PHOGs), color and edge directivity descriptor (CEDD, local binary patterns, and scalable color (SC). Figure 2 shows the architecture of the proposed GF-based methods (1 and 2). These methods use six selected GFs and the best ML classifiers for the provided dataset.

Feature engineering is among the most crucial and challenging parts for approaching any ML and computer vision problem. Based on the findings of Pogorelov et al. [54] and Riegler et al. [59], we choose to used JCD, Tamura, CL, EH, autocolor correlogram, and PHOG. The combinations of these features represent the overall properties of the images. We can even add more GFs, but doing so may increase the noise to the image features, which again would hurt the classification performance. Moreover, we have formulated the problem of GI tract anomaly classification as a multi-class (16-class) classification of different findings including anomalies, landmarks, and clinical markings. With the provided dataset, we computed the GFs of each image. A multi-class classification problem is a general and well-studied ML problem, and there is a variety of methods available to solve this issue with higher performance. Therefore, we sent the extracted GFs to many available ML classifiers. The whole experiment was completed with the development dataset. The 2018 Medico Task [53] shows the best classification rates with SimpleLogistic (SL) [38] and LMT [38] classifiers.

3.1 Method 1: The SL Classifier

In method 1, we combine the SL classifier from the Weka software [22] to build a linear logistic regression (LR) model with the LogitBoost [21] utility for determining attributes. The SimpleLogistic (SL) classifier can deal with binary class classification, multi-class classification, missing class, and nominal class. It can handle different types of attributes, such as binary attributes, nominal attributes, date attributes, missing values, unary attributes, and empty nominal attributes [38]. In a linear LR classifier, a simple (linear) model fits the data, and the method of model fitting is pretty stable, leading to low variances.

LogitBoost is utilized for determination of the most appropriate attributes in the data at the time of executing LR, which is done by performing a simple regression in every iteration before it converges to a solution of maximum likelihood. Therefore, LogitBoost, with a simple regression function that acts as a base learner, is utilized for fitting the logistic models. The optimum number of iterations associated with the LogitBoost algorithm to function is cross validated, which leads to the automatic selection of the attribute [65]. The SL classifier has a built-in attribute selection (if the default parameter is not changed): it stops computing simple linear regression models (i.e., performing LogitBoost iterations) when the cross-validated classification error no longer decreases. With the extracted features using LIRE, the SL classifier has not only the highest classification accuracy but also takes the lowest classification time (i.e., lowest computational complexity) when compared with other ML classification algorithms.

3.2 Method 2: The LMT

In method 2, we use the Logistic Model Tree (LMT) classifier from the Weka software. The LMT is a classification model related to a supervised training algorithm, which is a combination of LR and decision tree learning techniques [38, 62]. Thus, the LMT is considered an analogue model for solving classification problems. In the logistic variant, information gain is utilized for splitting, the LogitBoost algorithm generates an LR model at each node in the tree, and the CART algorithm [62] is utilized for pruning the tree.

The LMT uses a cross-validation (CV) technique to find several LogitBoost iterations to prevent overfitting of the training data. The LogitBoost algorithm accomplishes additive LR, which is achieved by least-square fits for every class M [19], which is shown in Equation (1):

$$L_M(x) = \sum_{i=1}^{n} \beta_i + \beta_0. \tag{1}$$

Here, β_i denotes the coefficient of the ith component of the vector x, and n denotes number of features. The LMT model uses the linear LR method to calculate the posterior probabilities of the leaf nodes [38], which is shown in Equation (2):

$$L_M(X) = -\frac{\exp\left(L_M(X)\right)}{\sum_{M=1}^{D} \exp\left(L_M(X)\right)}. \tag{2}$$

Here, D denotes the number of classes, and $\mathrm{L}_{\mathrm{M}}(\mathrm{X})$ stands for the least-square fits. The least-square fits $\mathrm{L}_{\mathrm{M}}(\mathrm{X})$ are transformed in such a way that $\sum_{M=1}^{D} \exp(L_M(X))\Big)$ is equal to zero.

4 DL APPROACHES

For our transfer learning approaches, we selected two DNNs: ResNet-152 [23] and DenseNet-161 [26] based on the top-1 error rate and top-5 error rate for the ImageNet [17, 61] classification as given in the PyTorch documentation [14]. Then, we chose ResNet-152 as the base model of the first DL approach, and this base model experiment was done under method 3 (the model is illustrated in Figure 3). This selection was made based on preliminary experiments. In the preliminary experiments, ResNet-152 showed better performance than DenseNet-161. This DenseNet-161 was in second place in the performance ranking when we compared stand-alone pre-trained DL models.

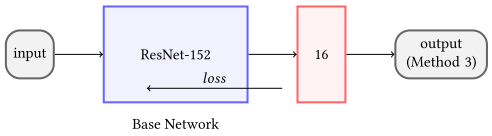


Fig. 3. Block diagram of method 3. The input is an image that is passed to a ResNet-152 neural network. A final softmax layer outputs the scores for the 16 classes.

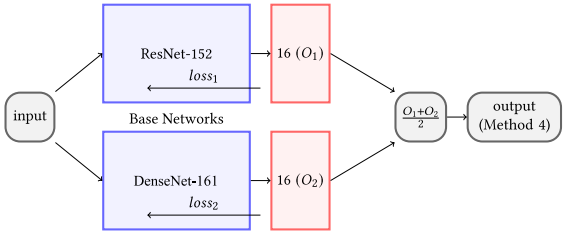


Fig. 4. Block diagram of method 4. The input image is in parallel passed to a ResNet-152 and a DenseNet-161 neural network. Two separate softmax layers calculate separate 16-class scores, which are finally combined.

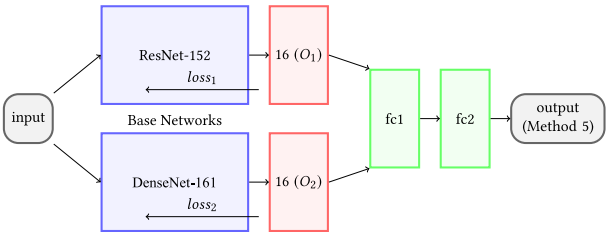


Fig. 5. Block diagram of method 5. It is similar to method 4, but instead of a single step to combine the output scores of the two neural networks, two fully connected layers are utilized.

In DL methods 4 and 5 (as illustrated in Figures 4 and 5), we used both pre-trained ResNet-152 and DenseNet-161 using the ImageNet dataset. In the following sections, we discuss data pre-processing mechanisms and training mechanisms used for all three DL methods. In later sections, we discuss these methods one by one with their fine-tuning mechanisms with more comprehensive explanations.

For the transfer learning methods, we use the data pre-processing tool of the PyTorch library to (i) resize input images, (ii) crop marginal annotations of the medical images, (iii) normalize the pixel values of input images, and (iv) apply random image transformations. Regarding image resizing, all images of the dataset were resized into 224×224 because ResNet-152 and DenseNet-161 accept images with these dimensions. By applying the central-cropping transformation of PyTorch, we minimized unnecessary effects for the final predictions of DNNs affected from annotated marks (green boxes) of the medical images as shown in Figure 1(b), (n), and (o). Center cropping did not remove important information from the images because we cropped down to 224×224 from 256×256. Our experiments show that removing the whole green box, such as those in Figure 1(b), (n), and (o), from the images by applying a larger crop size is not advisable, because for some images, too much content of the finding is lost with a large crop size. When applying the normalization function to the input images, a standard deviation (σ) of 0.5 and a mean (μ) of 0.5 were used with the normalization function in PyTorch. The mathematical equation used in this function is given in Equation (3), and c represents the three channels R, G, and B of input images. The *input* represents a tensor of pixel values of each layer. We used random transformations, random horizontal

flips, random vertical flips, and random rotations from PyTorch as data augmentation techniques.

$$input_c = \frac{input_c - \mu_c}{\sigma_c}; \quad where \quad c = [0, 1, 2] \tag{3}$$

For training all DNNs, the transfer learning mechanism was used. Then, we used cross-entropy loss [15] with weighted classes as given in Equation (4) to calculate the loss values of the DNNs:

$$loss(x, class) = weight[class] \times \left(-x[class] + \ln \left(\sum_j \exp(x[j]) \right) \right). \tag{4}$$

In this equation, the weight parameter value is calculated inversely proportional to the image count in the corresponding class. In other words, class weight values are high when the classes have fewer images. However, the inbuilt cross-entropy function given in PyTorch is used instead of implementing it from scratch. While doing preliminary experiments, we observed that there was not any effect from weighted cross-entropy loss. Then, we used the normal cross-entropy loss (Equation (5)) function for calculating the loss of the DNNs:

$$loss(x, class) = -\ln \left(\frac{\exp(x[class])}{\sum_j \exp(x[j])} \right) = -x[class] + \ln \left(\sum_j \exp\left(x[j]\right) \right). \tag{5}$$

As the optimizer of all DNNs, the stochastic gradient descent (SGD) [11] method with a momentum [66] was applied. We selected this optimizer because of its stable learning mechanism in contrast to the highly unstable learning pattern of other methods [35, 60, 80], as they show fast convergence.

During the training procedure, we changed the learning rate manually based on the progress of learning curves rather than using the inbuilt learning rate schedulers of PyTorch. Initially, we began with a high learning rate. Then, the learning rate was reduced by a factor of 10 if the training process did not show good progress in the learning curves. Finally, model weights of the best epoch based on the best validation accuracy were saved to use in the inference stage.

4.1 Method 3: DNN Approach Based on ResNet-152

Method 3 is the base method that uses only ResNet-152. A block diagram of this is illustrated in Figure 3. In this method, the last layer of ResNet-152 is modified to output 16 classes of the 2018 Medico Task from 1,000 classes of ImageNet. Usually, we freeze first layers (there is not a logical way to select the number of layers to freeze) of pre-trained networks when we do transfer learning. Then, we train the last and the new layers using the new domain data. Finally, the entire network is trained after unfreezing all parameters of the network (a method known as fine tuning).

We performed preliminary experiments to identify the influence of the preceding freezing-unfreezing technique compared to using simple fine tuning. Both techniques showed the same performance at the end of the training process, and we could not gain any performance benefit from the freezing-unfreezing method, as using the simple fine-tuning method was faster. Therefore, we decided to use the simple fine-tuning method for all experiments.

In method 3, we started the training process with a learning rate of 0.001. Then, the learning rate was decreased by a factor of 10 if we could not see any performance improvement for the validation dataset. We repeated this change of learning rate until the model came to a good stable position. In this experiment, the SGD method was used as the optimization method with a momentum of 0.9.

4.2 Method 4: DNN Approach Based on ResNet-152 and DenseNet-161

In method 4, as illustrated in Figure 4, we used two pre-trained networks on ImageNet: ResNet-152 and DenseNet-161. These networks were retrained separately into the Medico dataset using the same procedure used in

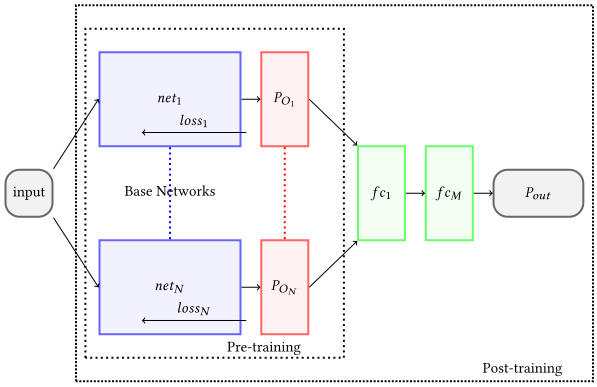


Fig. 6. Block diagram of the proposed parallel DNN merging. The training process is split into a pre-training (pre-training of individual models) and post-training step (training the whole network architecture).

method 3. Before this retraining, the networks were modified to classify the 16 classes. Then, we calculated an average probability of the two probability vectors (V_{Resnet_152} and $V_{Densenet\_161}$) output by the two separate networks: ResNet-152 and DenseNet-161. By calculating the average of these two probability vectors ($V_{answer} = \frac{1}{2}\,(V_{Resnet\_152} + V_{Densenet\_161})$, we accepted the cumulative probability decision rather than the individual decision. Using the average from these two networks, we expected to have a good decision with high confidence. For example, if the two networks return high probability values for the same class, the class probability value (confidence of classifying to that class) is high. However, when one network has a high probability and the other network has a low probability for a specific class, then the final probability value is around 0.5. This value infers that confidence about the particular class is not good enough for the final decision.

In this model, the probability of the final answer depends on the average values rather than the highest probability value returned from one of the two models. Here, the problem is that the prediction suggested from the highest probability value of one model may be the correct class compared to the selected category from the average. Finally, we trained the model using a learning rate of 0.001. In addition, we decreased the learning rate by a factor of 10 when the model did not show convergence. A momentum of 0.9 with SGD was used as in method 3.

4.3 Method 5: DNN Approach Based on ResNet-152, DenseNet-161, and MLP

Method 5 was designed to overcome the problem of method 4. The block diagram of this method is illustrated in Figure 5. The simple averaging method was not enough to make a final decision when the two networks provided two different answers. As a solution, an Multi-Layer Perceptron (MLP) was introduced instead of the simple averaging method. Then, we trained only this MLP with the pre-trained ResNet-152 and DenseNet-161 for the Medico dataset to decide the final prediction based on the probabilities that come from two networks. More details about designing this complex model are discussed in Sections 4.3.1 and 4.3.2.

4.3.1 Extendable Method 5. In this section, we show how we can improve accuracy using multiple cumulative probabilistic decisions by extending method 5 into $N \geq 2$ DNNs. In this general model, as illustrated in Figure 6, we divide the whole training process into the following four steps: (1) pre-training of individual models, (2) model selection for merging, (3) merging models with an MLP, and (4) post-training and fine tuning. Let $NETS = \{net_1, net_2, \ldots, net_N\}$ be the set of pre-trainer networks using the ImageNet dataset and P_{O_i} be the returned probability vector for model net_i.

In step 1 (pretraining), we train each DNN $net_i \in NETS$ as much as possible using the transfer learning mechanism until it gives the best predictions as described in method 3 (using different loss functions; $loss_1$ to $loss_N$).

The DNNs have their unique prediction capabilities within the given classification problem. Then, we analyze the CM of the best outcome of each DNN.

In step 2 (selection), we select networks that give different diagonals of CMs (the diagonal of a CM represents correct classifications) compared to other CMs of selected DNNs. If the diagonal of CM of network $net_i = CM_i$, then we select networks that have $CM_i \neq CM_j; j = [1, 2, \ldots, i-1, i+1, \ldots, N]$. The goal of this comparison is to identify DNN models that have different classification performances compared to each other. Equal diagonals of CMs do not imply that the networks are identical for their classifications, because there might be models that give the same diagonal numbers but lead to different classifications for a given image. If the case of equal diagonals occurs, we have to compare correctly classified images to identify the differences. The number of DNNs selected for the final training may or may not be equal to the initial number of pre-trained DNNs depending on similarities in some of the CMs.

In step 3 (merging), we use an MLP to merge all outputs of the selected DNNs. The MLP consists of M layers that take $\sum_{i=1}^{N} length\_of(P_{O_i})$ number of inputs and output P_{out} probability vector according to the given classification problem. Then, step 4 can be started by freezing all the pre-trained DNNs and training only the new MLP until it shows a good validation performance. Optionally, we can retrain the whole model without freezing any layer if we cannot achieve a performance improvement by training only the new MLP.

4.3.2 Method 5 Used by This Research Work. According to the procedure discussed in Section 4.3.1, our implementations of method 5 were designed using two parallel networks ($N = 2$): ResNet-152 and DenseNet-161. Then, we analyzed two CMs, which came from ResNet-152 and DenseNet-161. These two networks were pre-trained according to the given classification problem. Because $CM_{Resnet\_152} \neq CM_{Densenet\_161}$, we combined the two networks with an MLP. This comparison of CMs was done visually using colormaps. However, if the visual inspection of CMs is hard, mathematical operations can be used. Moreover, if the CMs are equal completely, a manual inspection of the classified images is required to identify the differences of model classifications. After combining, we froze two DNNs to proceed to the post-training step. In our experiments, the input layer of the MLP consisted of 32 input nodes. The output of the MLP was a probability vector with 16 values, which is equal to the number of classes of the Medico dataset. We used two fully connected layers, with 32 neurons and 16 neurons. In the post-training step, we started training only the MLP with a learning rate of 0.01. To do the post-training, multi-class cross-entropy loss and Stochastic Gradient Descent (SGD) were used.

5 RESULTS

In this section, we discuss the experimental setup, datasets, and results obtained from our experiments. Using these presented results, we emphasize that high scores for performance metrics do not always show the actual performance of ML methods. To show this, we present well-performing ML models that achieved good results for their performance values. Using cross-dataset testing, we present a detailed analysis of evaluation metrics to emphasize that they are not always representative to identify the real performance of models.

For all experiments, we used the same hardware platform with an Intel Core i7 eighth-generation processor with 16 GB of DDR4 RAM and an 8-GB NVIDIA GeForce 1080 GPU. However, we practiced two different software frameworks for implementing our methods. To implement the GF-based methods (1 and 2), we used the Weka framework [22]. We used the PyTorch framework for the DNN-based methods (3, 4, and 5).

5.1 Datasets

For the work performed in this article, we used the following four datasets: the 2018 Medico dataset [55], CVC-356-plus (a modified version of CVC-356 [6, 7, 73]), CVC-612-plus (a modified version of CVC-612 [6, 7, 73]), and CVC-12k [4, 5]. The training and testing datasets of the 2018 Medico Task were derived from the Kvasir dataset [51] and Nerthus dataset [50], consisting of 16 classes as shown in Table 2. These images consist of different anatomical landmarks (z-line, pylorus, cecum), pathological findings (esophagitis, polyps, ulcerative

Table 2. Summary of the 2018 Medico Dataset

Type	Images in the Development Set (#)	Images in the Test Set (#)
Blurry-nothing	176	39
Colon-clear	267	1,070
Dyed-lifted-polyps	457	590
Dyed-resection-margins	416	583
Esophagitis	444	483
Instruments	36	165
Normal-cecum	416	604
Normal-pylorus	439	569
Normal-z-line	437	636
Out-of-patient	4	6
Polyps	613	423
Retroflex-rectum	237	194
Retroflex-stomach	398	399
Stool-inclusions	130	508
Stool-plenty	366	1,920
Ulcerative-colitis	457	551

The first column shows the names of the different findings. The second and third columns show the number of images in the development and test sets.

Table 3. Overview of the Datasets Used for Our Experiments

Dataset	Training	Testing	Images (#)	Polyps (#)	Non-Polyps (#)
2018 Medico—Development	X	—	5,906	613	5,293
2018 Medico—Testing	—	X	8,740	423	8,317
CVC-356-plus	X	X	2,285	356	1,929*
CVC-612-plus	X	X	1,316	612	704
CVC-12k	—	X	11,954	10,025	1,929

*We replaced this image set with a new image set (with 1,171 images) extracted from a clear colon video collected from the Bærum Hospital, Norway, in the second stage of this research to avoid the overlap between the training data and the testing data.

In total, we have five different datasets, but the Medico dataset is split into a development part and a test part for the challenge. The training and testing columns indicate how the dataset was used in the experiments. Polyps and non-polyps indicate the number of findings. Medico and CVC-356-plus represent a bias toward non-findings. CVC-612-plus is a quite balanced dataset, and CVC-12k presents a bias toward findings. Datasets were chosen based on these distributions to represent common cases in medical imaging datasets.

colitis), endoscopic polyp removal cases (dyed and lifted polyp, dyed resection margin), and normal findings (normal colon mucosa, stool) in the GI tract. The dataset also contains images with different degrees of the Boston Bowel Preparation Scale (BBPS), ranging from 0 to 3. Some of the original images contain the endoscope position marking probe. These are seen as a small green box located in the bottom corners, showing its configuration and location of the image frame. The images used in the study were captured using an electromagnetic imaging system (Scopeguide, Olympus, Europe) [51]. In Table 3, we present a summary of the uses of the 2018 Medico dataset and other datasets for polyp and non-polyp classifications.

The Medico development dataset was used to train our ML models in the first stage. However, this dataset consists of a highly imbalanced number of images, as summarized in Table 2. Within this, the out-of-patient class had only 4 images to train our models. Therefore, only in the first stage, we used an additional 30 images

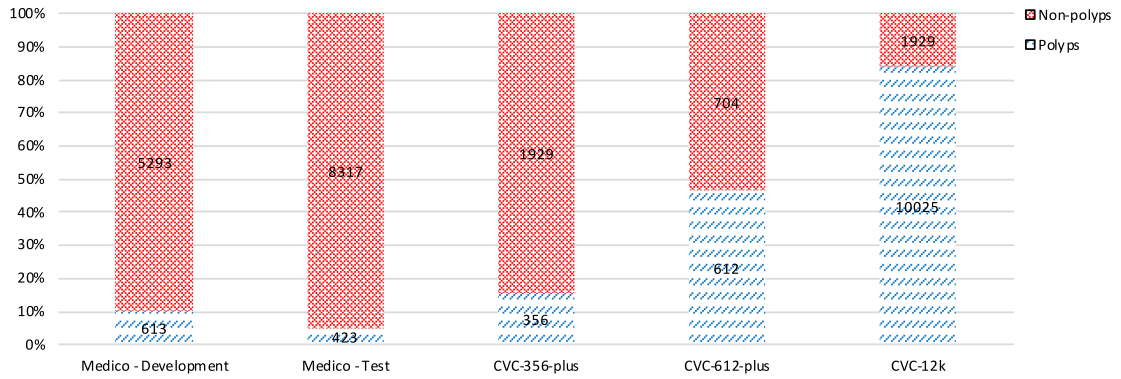


Fig. 7. Ratios of findings to non-findings in the datasets (polyp/non-polyp). The *X* axis represents the different datasets used for the binary classification. The *Y* axis represents the percentage of polyps and non-polyps. The numbers inside the bars show the actual number of polyp and non-polyp images.

that were selected randomly from the Internet to fill this class in the training dataset. These were images of flowers, vehicles, and other general stuff in our everyday life and did not have any relationship with this class. The advantage of this technique is discussed in the discussion of Section 6.

When we discussed the ML models' generalizability in the second part of the article, we used the CVC datasets to retrain and test our models. The CVC-356-plus dataset is the modified version of the CVC-356 [6, 7, 73] dataset that has only polyp images. In that modification, we added 1,929 non-polyp images from the CVC-12k [4, 5] dataset to the CVC-356 dataset and created a new dataset called *CVC-356-plus*. Similarly, the CVC-612-plus dataset was created by extending the CVC-612 dataset [6, 7, 73]. For this CVC-612-plus dataset, we added 704 non-polyp images extracted from new GI tract videos collected by the Bærum Hospital, which is part of the Vestre Viken Hospital Trust in Norway. The content of the CVC-12k dataset underwent a minor reorganization by filtering and grouping polyp and non-polyp images into two separate folders. However, the content and number of images in CVC-12k were not otherwise changed. Therefore, we refer to it by its common name.

In the second part of our research, we used the CVC-356-plus and CVC-612-plus datasets for retraining our models to classify polyps and non-polyps. In only this part of the research, we replaced 1,929 non-polyp images of the CVC-356-plus dataset with 1,171 newly extracted images from a clean and healthy colon video collected from the same hospital. We did this modification to avoid the overlap between the non-polyp images of the CVC-356-plus training dataset and the CVC-12k testing dataset.

For the dataset preparation stage, we focused on the number of polyp and non-polyp images in each dataset to analyze the correlation between the data distribution and the model performance. A bar graph of this data distribution is illustrated in Figure 7. We chose to include different proportions for the number of polyps and non-polyps to keep diversity of data percentages in each test case. In the CVC-356-plus dataset, the polyp percentage is low compared to the non-polyp percentage. In the CVC-612-plus dataset, percentages of polyps and non-polyps are around 50%. In contrast, the CVC-12k dataset has a higher polyp percentage than the non-polyp percentage. Due to this, we can study the effects of data imbalance in the training and testing datasets on the performance and interpretability of the metrics.

5.2 Analyzing Results

We discuss our results in two main sections: (i) the 16-class classification task based on the 2018 Medico Task and (ii) the polyp and non-polyp classification task to analyze generalizability of ML models.

Table 4. Evaluation Results of the 2018 Medico Task (as Provided by the Organizers of the 2018 Medico Task) [71] for the Five Methods Used in This Article

Method	REC	PREC	SPEC	ACC	MCC	F1
1	0.8457	0.8457	0.9897	0.9807	0.8353	0.8456
2	0.8457	0.8457	0.9897	0.9807	0.8350	0.8457
3	0.9376	0.9376	0.9958	0.9922	0.9335	0.9376
4	0.9400	0.9400	0.9960	0.9925	0.9360	0.9400
5	0.9458	0.9458	0.9964	0.9932	0.9421	0.9458

Based on the official results, method 5 was the best one based on the MCC score.

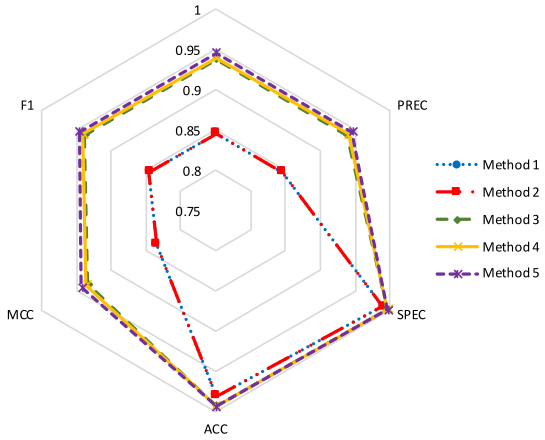


Fig. 8. Performance comparison of all five classification models for the 16 classes of the 2018 Medico test dataset. Methods 1 and 2 are similar in results but different from the other three methods (note that measurements start at 0.75).

5.2.1 16-Class Classification. In this 16-class classification task, the training dataset of the 2018 Medico Task was split into a 70% training dataset and a 30% validation dataset. Then, the test data given by the organizers was used to test the performance of five methods for classifying 16 classes of the GI tract findings.

We evaluated our five models based on the results collected by the organizers. The evaluated results of the main five models are tabulated in Table 4. With an MCC score of 0.9421, method 5 showed the best performance for classifying the 16 classes of GI tract findings. However, our GF-based approaches did not show results competitive with the DNN methods. The GF model introduced in method 1 could reach an MCC score of 0.8353. This result showed the best performance record for a GF-based method. A clear performance difference between the GF-based methods and the DNN-based methods can be seen in Figure 8. In this plot, we compared this performance difference using six performance measures: recall (REC), precision (PREC), specificity (SPEC), accuracy (ACC), MCC, and F-score (F1). According to this plot, it is clear that the areas of the hexagons covered by the GF methods are smaller than the areas covered by DNN methods. These results imply that three DL methods outperform two GF methods.

The CM of method 5 collected from the organizers of the 2018 Medico Task is tabulated in Table 5 for the in-depth investigation. According to the CM, we can identify two main bottlenecks to improve the performance of method 5. The first one is misclassification between esophagitis and normal-z-line, and the second one is misclassification between dyed-lifted-polyps and dyed-resection-margins. Therefore, images from these classes were manually examined to identify the reasons for these misclassifications. For the conflict between esophagitis

Table 5. CM of Method 5 (Our Best Model) Based on the Medico Test Dataset

		Actual Class															
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Predicted Class	Ulcerative-colitis (A)	500	—	—	—	–	—	—	—	39	—	3	—	1	1	—	7
	Esophagitis (B)	3	432	48	—	—	—	—	—	—	—	—	—	—	—	—	—
	Normal-z-line (C)	1	121	513	—	—	—	—	—	—	—	—	—	1	—	—	—
	Dyed-lifted-polyps (D)	1	—	—	522	31	—	—	—	—	—	2	—	—	—	—	34
	Dyed-resection-margins (E)	—	—	—	33	532	—	—	—	—	—	1	—	—	—	—	17
	Out-of-patient (F)	—	—	—	—	1	5	—	—	—	—	—	—	—	—	—	—
	Normal-pylorus (G)	3	3	2	—	—	—	559	—	—	—	2	—	—	—	—	—
	Stool-inclusions (H)	—	—	—	—	—	—	—	501	7	—	—	—	—	—	—	—
	Stool-plenty (I)	1	—	—	—	—	—	—	—	1,918	—	—	—	—	—	—	1
	Blurry-nothing (J)	1	—	—	—	—	—	—	—	1	37	—	—	—	—	—	—
	Polyps (K)	10	—	—	1	—	—	1	—	—	—	358	6	—	1	—	46
	Normal-cecum (L)	18	—	—	—	—	—	—	—	—	—	6	578	—	—	—	2
	Colon-clear (M)	1	—	—	—	—	—	—	5	—	—	—	—	1,063	—	1	—
	Retroflex-rectum (N)	3	—	—	—	—	—	—	—	—	—	2	—	—	188	1	—
	Retroflex-stomach (O)	—	—	—	—	—	—	1	—	—	—	—	—	—	2	395	1
	Instruments (P)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	165

The diagonal value represents true predictions (number of images) of the model. A through P are the classes corresponding to the class names in the first column. The most confusion can be observed between classes B and C, and classes D and E. Looking at the images, we can see that they are quite similar in their visual features (colors, texture, etc.).

and normal-z-line, the reason is the very close locations of these two landmarks in the GI tract. However, the confusion between dyed-lifted-polyps and dyed-restrictions is caused because of the same color patterns and the same texture structures of both types of images. With these limitations, method 5 showed the best performance with an MCC of 0.9421, which was the important measurement to win the 2018 Medico Task. Based on the MCC value, we won second place in the 2018 Medico Task. The winning team [25] relabeled the development dataset and also generated more images out of the provided instruments class by placing the instrument as a foreground over the images of dyed-lifted-polyps, dyed-resection-margins, and ulcerative colitis to balance the instrument class for improving performance. However, we developed the model by only using the images provided by the task organizers for a fair comparison of the approaches with the limited dataset. Then, our next experiments were conducted to find the reusability of these well-performed models in different datasets with polyp and non-polyp categories (sub-categories of the 16 classes of primary tasks).

5.2.2 Polyp and Non-Polyp Classification Using the Pre-Trained Models. The following analysis was performed to identify the polyp classification ability of our five models on the same test dataset and different CVC datasets. The 16-class classification results collected from the Medico Task organizers were analyzed to calculate polyp detection performance in the Medico test data. Moreover, our models were tested with the CVC-356-plus, CVC-612-plus, and CVC-12k datasets without any modifications to the five models to compare the performance of polyp detection.

According to the correct and incorrect classifications of polyps and non-polyps in the test datasets, the first large column of Table 6 was calculated to measure the polyp detection performance of five models. In this evaluation process, all 15 classes except the polyp class were considered as the non-polyp classification because the number of outputs is 16 in the first models. For comparison, the MCC values of these tests are plotted in Figure 9. This graph shows that the polyp detection performance of the same dataset (the testing dataset of the Medico Task) is higher than on the completely new datasets (CVC-356-plus, CVC-612-plus, and CVC-12k) for both the

Table 6. Polyp Classification Results with and without Retraining for All Datasets and Methods

		Without Retraining						With Retraining to 2-Class Classification					
	M	REC	PREC	SPEC	ACC	MCC	F1	REC	PREC	SPEC	ACC	MCC	F1
Test Dataset	1	0.7834	0.4899	0.9635	0.9558	0.5987	0.6028	0.9550	0.9630	0.6740	0.9553	0.5430	0.9590
	2	0.7834	0.4899	0.9635	0.9558	0.5987	0.6028	0.9540	0.9630	0.6840	0.9537	0.5400	0.9580
	3	0.9733	0.8088	0.9897	0.9890	0.8819	0.8835	0.9813	0.6577	0.9772	0.9773	0.7934	0.7876
	4	0.9599	0.8467	0.9922	0.9908	0.8969	0.8997	0.9813	0.7384	0.9845	0.9843	0.8440	0.8427
	5	0.9572	0.8463	0.9922	0.9907	0.8954	0.8984	0.9706	0.7516	0.9857	0.9850	0.8470	0.8471
CVC-356-plus	1	0.3089	0.1053	0.5158	0.4835	−0.127	0.1571	0.8450	0.7990	0.1700	0.8446	0.0750	0.7780
	2	0.3089	0.1053	0.5158	0.4835	−0.127	0.1571	0.8510	0.8420	0.2070	0.8512	0.1930	0.7930
	3	0.7865	0.3738	0.7569	0.7615	0.4198	0.5068	0.8118	0.5547	0.8797	0.8691	0.5978	0.6591
	4	0.6713	0.4003	0.8144	0.7921	0.4010	0.5016	0.6517	0.4150	0.8305	0.8026	0.4068	0.5071
	5	0.6685	0.4837	0.8683	0.8372	0.4737	0.5613	0.6713	0.6408	0.9305	0.8902	0.5906	0.6557
CVC-612-plus	1	0.7696	0.7969	0.8295	0.8016	0.6008	0.7830	0.6980	0.8070	0.6530	0.6983	0.4740	0.6590
	2	0.7696	0.7969	0.8295	0.8016	0.6008	0.7830	0.7220	0.8170	0.6800	0.7218	0.5140	0.6910
	3	0.8415	0.6242	0.5597	0.6907	0.4137	0.7168	0.8382	0.6136	0.5412	0.6793	0.3932	0.7086
	4	0.8627	0.6559	0.6065	0.7257	0.4803	0.7452	0.8578	0.6890	0.6634	0.7538	0.5265	0.7642
	5	0.8137	0.6501	0.6193	0.7097	0.4379	0.7228	0.8007	0.7061	0.7102	0.7523	0.5104	0.7504
CVC-12k	1	0.4858	0.8391	0.5158	0.4907	0.0012	0.6154	0.1650	0.7880	0.8370	0.1651	0.0130	0.0530
	2	0.4858	0.8391	0.5158	0.4907	0.0012	0.6154	0.1650	0.8210	0.8380	0.1699	0.0290	0.0630
	3	0.6112	0.9289	0.7569	0.6347	0.2722	0.7373	0.6033	0.9631	0.8797	0.6479	0.3558	0.7419
	4	0.6236	0.9458	0.8144	0.6544	0.3241	0.7517	0.6459	0.9519	0.8305	0.6757	0.3539	0.7696
	5	0.5936	0.9591	0.8683	0.6379	0.3401	0.7333	0.5576	0.9766	0.9305	0.6178	0.3595	0.7099

M, method.
For training, 2018 Medico development data was used. We can observe that for some datasets, retraining seems to improve performance.

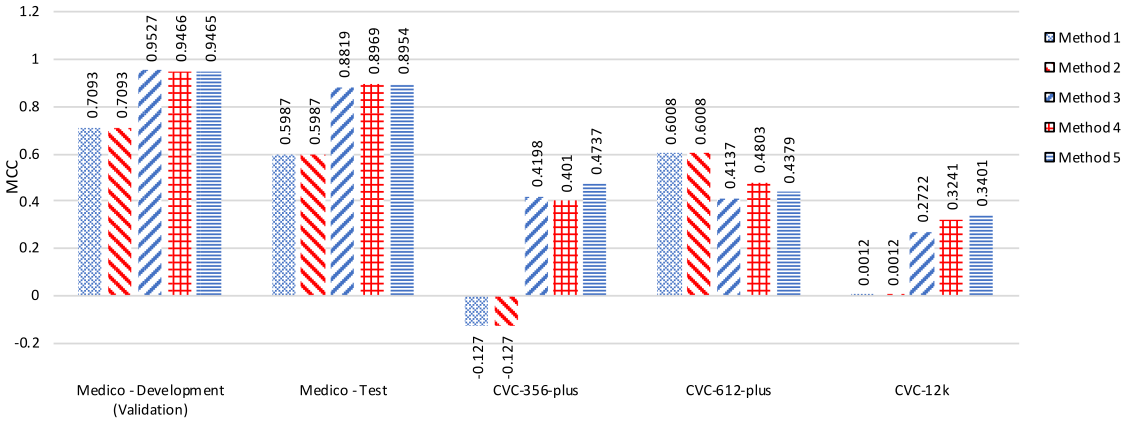


Fig. 9. Polyp and non-polyp classification capabilities (based on MCC) of all five methods that were trained using 2018 Medico development data to classify 16 classes. For most cases, methods 3 through 5 perform best. For the CVC-612-plus test data, methods 1 and 2 perform best.

GF-based approaches and the DNN approaches. This is the first analysis, and we emphasize that it shows that researchers need to do cross-dataset evaluations to prove the real capabilities of ML models.

From the first column of Table 6 and Figure 9, it is clear that the performance of the GF methods for different datasets (CVC-356plus, CVC-612-plus, and CVC-(356+612) dataset) is unpredictable because it presents huge value fluctuations in the graph with a negative MCC value. This shows the incapability of GF methods to make predictions on different datasets. The negative values of MCC in this experiment, such as −0.127 for the CVC-356-plus dataset, indicate that there is no agreement or only a non-relevant relationship between target and prediction. An MCC around zero would mean that the classifier is deciding randomly, and MCCs above zero would indicate correct classification. The closer to −1 or 1, the stronger the indication for being wrong or correct, respectively. However, the polyp detection performance of the GF-based methods in the CVC-612-plus dataset outperforms the DNN methods with an MCC value of 0.6008, whereas the best DNN method shows an MCC value of 0.4803. This prediction accuracy of the GF methods can be identified as an erroneous prediction, because the performance of this method for the other two CVC datasets shows poorer MCC scores than those of DNN-based approaches. Moreover, the DNN-based approaches show considerable steady MCC values for all new datasets, implying that the DNN methods are more generalizable than the GF methods.

Because the performance gap between the 16-class classification and polyp classification showed differences, we retrained our models to classify only the polyp and non-polyp classes. Therefore, our next experiments were performed to test how retraining our five ML models to classify polyps and non-polyps will influence performance.

For the retraining experiments, we first retrained the two GF methods with new ARFF files generated for polyp and non-polyp categories. Second, in the retraining stage of the three DNN methods, we changed only the last layer into two outputs. However, we did not change the loss function from categorical cross-entropy into binary cross-entropy because two-class categorical cross-entropy is equal to binary cross-entropy. Moreover, we retained the original optimization functions. Then, we retrained all five models using the same Medico dataset, which has only polyp and non-polyp classes. The results of these experiments are tabulated in the right columns of Table 6.

The results in Table 6 show that it can be difficult to evaluate the models and interpret the results after retraining for two-class classification. All MCC values of the five methods tested on the CVC-356-plus data show improvements. Similarly, for the CVC-612-plus test data, methods 4 and 5 show performance improvements from MCC values of 0.4803 and 0.4379 to 0.5265 and 0.5104, respectively. In contrast, methods 1, 2, and 3 show a performance drop, which is indicated by MCC values 0.6008, 0.6008, and 0.4137 reduced to 0.4740, 0.5140, and, 0.3932, respectively. Therefore, we extended our experiment by introducing additional retraining options with the CVC-356-plus and CVC-612-plus datasets. After that, the retraining process can be categorized as retraining the models to classify polyps and non-polyps using (i) only the same Medico training dataset (as tabulated in Table 6), (ii) the Medico dataset with the CVC-356-plus dataset, (iii) the Medico dataset with the CVC-612-plus dataset, and (iv) the Medico dataset with the CVC-356-plus and CVC-612-plus datasets. Then, our testing datasets are limited to two datasets: the Medico test dataset and the CVC-12k dataset. Results related to these new retraining processes can be seen in Table 7. When the models are trained using the balanced CVC-612-plus dataset in combination with the 2018 Medico development data, the DNN models show better MCC values (0.8189, 0.8555, and 0.8606) for methods 3, 4, and 5, respectively. This is true for the Medico test data and the two smaller CVC datasets. Moreover, the MCC values for the CVC-12k test data also achieve the best MCC values of 0.1421, 0.1418, and 0.1802 for methods 3, 4, and 5. An important observation from the CVC-12k dataset is also that looking at all other metrics but MCC and specificity could mislead to the assumption that the results are good—for example, scores above 0.8 for accuracy, which is often used as the only indicator for performance in similar studies.

In the first comparison, we plotted performance changes for the retraining with the different training datasets and tested them on the Medico test dataset. The changes in the Recall (REC), Precision (PREC), Specificity (SPEC), Accuracy (ACC), MCC, and F-score (F1) values can be seen as hexagon plots in Figure 10(a), (c), (e), (g), and (i),

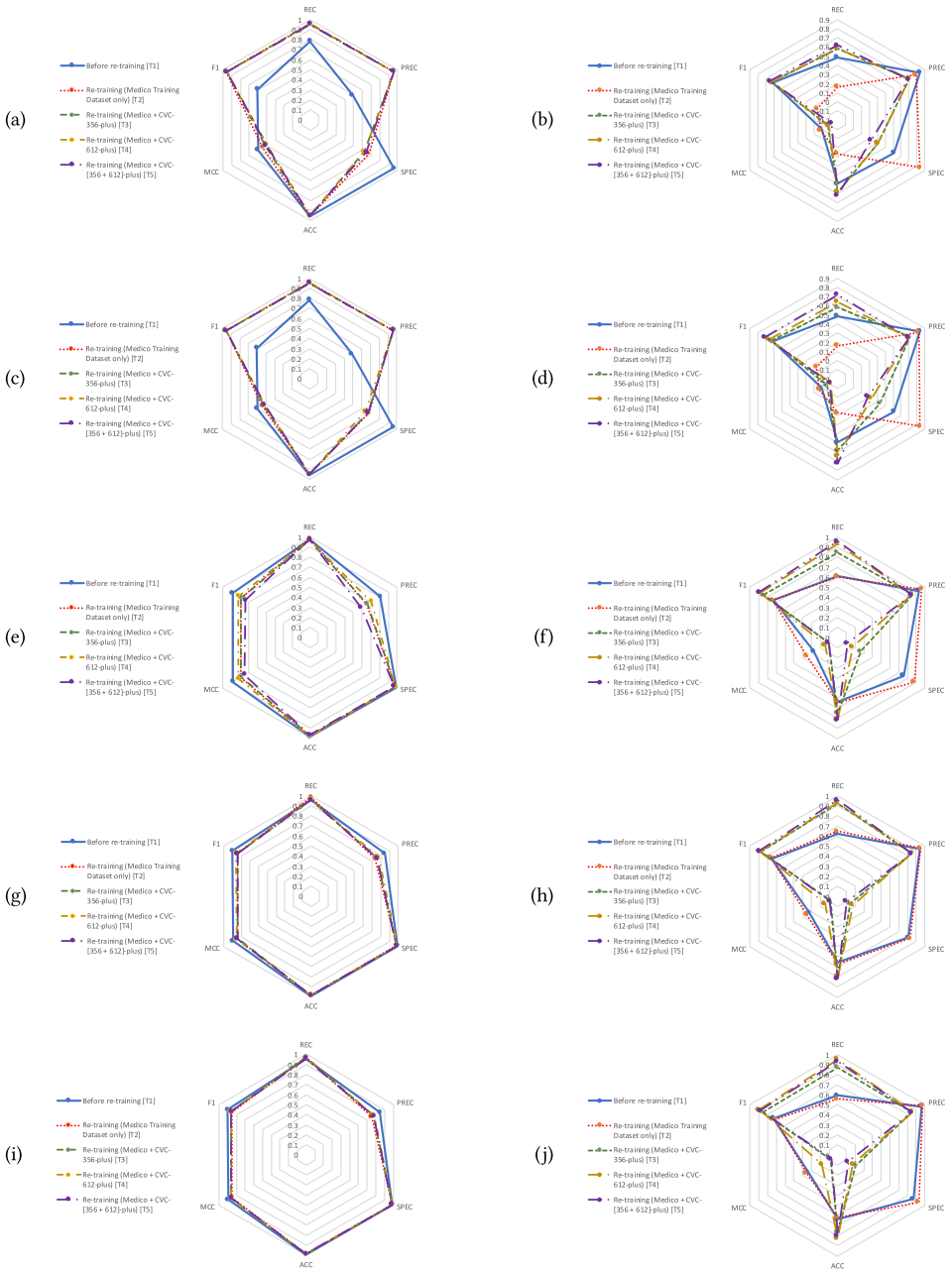


Fig. 10. Polyp and non-polyp classification using the proposed ML methods: 1, 2, 3, 4, and 5. The first column (sub-figures (a, c, e, g, i)) shows the results of the Medico test dataset, and the second column (sub-figures (b, d, f, h, j)) shows the results of the CVC-12k dataset. The methods are represented as follows: (a) and (b) for method 1, (c) and (d) for method 2, (e) and (f) for method 3, (g) and (h) for method 4, and (i) and (j) for method 5, respectively.

Table 7. Evaluation Results on Using CVC-356-plus and CVC-612-plus Combined as Training Data with Retraining to Classify Polyps and Non-Polyps

			MedicoTest Data						CVC-12k					
		M	REC	PREC	SPEC	ACC	MCC	F1	REC	PREC	SPEC	ACC	MCC	F1
Retraining Datasets with Medico Data	CVC-356-plus	1	0.9550	0.9610	0.6230	0.9549	0.5160	0.9570	0.5840	0.7040	0.3090	0.5836	−0.084	0.6320
		2	0.9520	0.9620	0.6710	0.9521	0.5260	0.9560	0.5810	0.7100	0.3360	0.5807	−0.065	0.6310
		3	0.9626	0.6630	0.9781	0.9775	0.7887	0.7852	0.8423	0.8565	0.2665	0.7494	0.1052	0.8493
		4	0.9599	0.7526	0.9859	0.9848	0.8427	0.8437	0.9192	0.8481	0.1441	0.7941	0.0810	0.8822
		5	0.9706	0.7773	0.9876	0.9868	0.8623	0.8633	0.8694	0.8507	0.2068	0.7625	0.0802	0.8599
	CVC-612-plus	1	0.9510	0.9590	0.6270	0.9508	0.4970	0.9540	0.5840	0.7030	0.3040	0.5842	−0.087	0.6320
		2	0.9530	0.9610	0.6430	0.9530	0.5160	0.9560	0.6400	0.6970	0.2240	0.6395	−0.117	0.6660
		3	0.9652	0.7092	0.9823	0.9816	0.8189	0.8177	0.9325	0.8546	0.1752	0.8103	0.1421	0.8918
		4	0.9572	0.7766	0.9877	0.9864	0.8555	0.8575	0.9336	0.8544	0.1731	0.8109	0.1418	0.8922
		5	0.9626	0.7809	0.9879	0.9868	0.8606	0.8623	0.9486	0.8571	0.1778	0.8242	0.1802	0.9005
	CVC-{356+612}	1	0.9500	0.9600	0.6480	0.9503	0.5050	0.9540	0.6180	0.6930	0.2280	0.6179	−0.129	0.6520
		2	0.9500	0.9610	0.6710	0.9503	0.5170	0.9550	0.7200	0.7010	0.1820	0.7199	−0.105	0.7100
		3	0.9733	0.5909	0.9699	0.9700	0.7458	0.7354	0.9537	0.8479	0.1109	0.8177	0.1028	0.8977
		4	0.9545	0.7596	0.9865	0.9851	0.8443	0.8460	0.9543	0.8463	0.0995	0.8164	0.0874	0.8971
		5	0.9599	0.7771	0.9877	0.9865	0.8571	0.8589	0.9278	0.8462	0.1239	0.7981	0.0699	0.8851

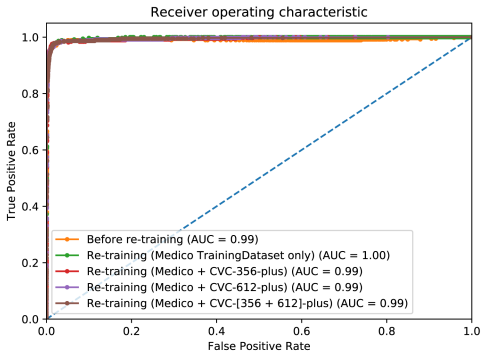
The 2018 Medico test dataset and the CVC-12k dataset are the test datasets. Using the balanced CVC-612-plus as training data, we achieve the best results. Combining CVC-356-plus and the CVC-612-plus does not improve performance. Overall, the performance is better on the Medico test dataset.

which correspond to methods 1, 2 ,3, 4, and 5, respectively. In these plots, T1 is used to present performance values before retraining the ML models into 2-class classification (binary classification). In this case, 15 classes except for the polyp class of the 16 classes were considered as the non-polyp class, and the polyp class is counted as the same polyp class. Furthermore, from T2 to T5, lines are used to present models with only two outputs. The T2 plot represents models' performance for the retraining using the Medico training dataset. Similarly, T3, T4, and T5 represent the retraining process using the Medico dataset and the CVC-356-plus dataset, the Medico dataset, and the CVC-612-plus dataset, and the Medico dataset, the CVC-356-plus dataset, and the CVC-612 dataset, respectively.

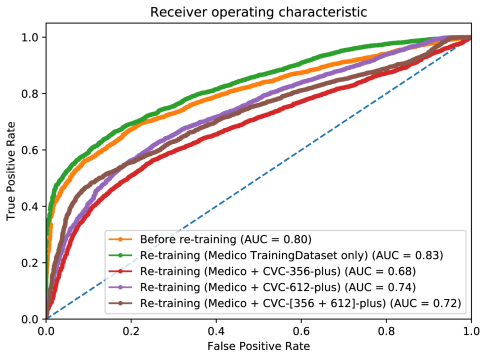
In the second series of experiments in this session, the same experiments were performed and tested on the CVC-12k dataset. The results obtained from these experiments are tabulated in Tables 6 and 7. Then, relevant results from these tables are plotted in Figure 10(b), (d), (f), (h), and (j). These plots use line notations similar to the preceding experiments.

Using the plot series in Figure 10, we can examine the reusability of ML models to classify polyps and non-polyps, which are sub-classes of the primary classes on the task. For example, if we compare plots in Figure 10(a) and (b), then we can know how method 1 performs to classify polyps and non-polyps within the test dataset the same as the training dataset and within an entirely new dataset. While investigating these plots, the proportion of the number of polyps and non-polyps is an important factor in explaining the shape of these hexagon plots.

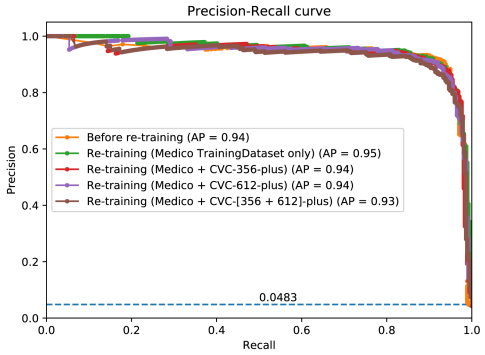
If we compare the GF methods (Figures 10(a) through (d)) and the DL methods (Figures 10(e) through (j)), it is clear that the DL methods outperform the GF methods in both the Medico Task and polyp classification task introduced in this article. This implies that the DL methods are capable of extracting deep features that cannot be extracted by manual feature extraction methods used by the GF methods. With the retraining process in the GF methods, we can see performance differences between the Medico dataset and the CVC-12k dataset. The main conclusion that we make is that GF-based methods are not able to capture the underlying patterns that would allow for efficient classification; thus, their performance is low.



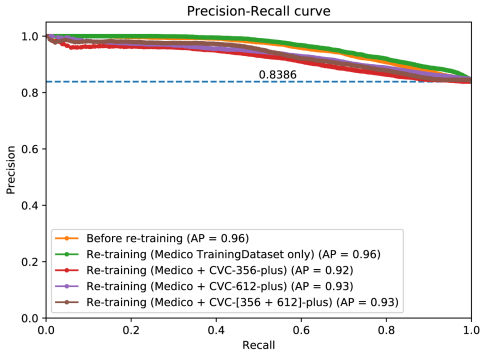
(a) ROC and PR curves for the Medico test dataset.



(b) ROC and PR curves for the CVC-12k dataset.



(c) PRC curves for the Medico test dataset.



(d) PRC curves for the CVC-12k dataset.

Fig. 11. ROC and Precision-Recall Curve (PRC) curves for method 5 trained on the CVC-356-plus and CVC-612-plus datasets as mentioned in the legends. Testing datasets are the CVC-12k and Medico test datasets. Overall, good performance can be observed in both ROC and PR curves. For CVC-12k, the PR curve shows the interesting case of a high random baseline for a biased dataset.

Plots in the first and second columns in Figure 10 show completely different behaviors for the same retraining process when we use different test datasets. The test dataset for the first column comes from the same domain as the training data, and the test dataset for the second column comes from the completely new domain, such as the CVC-12k dataset. To investigate these unusual performance changes, we generated and examined ROC and PR curves for the best DNN model (method 5). The ROC and PR curves for method 5 with the Medico test data (for the plot in Figure 10) are depicted in Figure 11(a) and (c). Similarly, the ROC and PR curves for method 5 with CVC-12k data (for the plot in Figure 10) are plotted in Figure 11(b) and (d).

Analysis of ROC curves is more robust for ML models that are used with balanced datasets, whereas PR curves are more valuable for ML methods when the methods engage with imbalanced datasets. However, we have used both curves in this paper to investigate the behavior of these curves while we are using highly imbalanced datasets. Consequently, the PR curves show completely different baseline values of 0.0483 for the Medico test dataset and 0.8386 for the CVC-12k dataset. The small baseline value arises in the plot in Figure 11(c) as a result of small polyps to the non-polyp proportion in the Medico test dataset. Conversely, the high baseline value in Figure 11(d) appears there as an effect on a high ratio of polyps to non-polyps.

To get a better understanding of the above plots, we selected the plots in Figure 10(i) and (j), and ROC and PR curves in Figure 11. With this selection, first, we analyzed T1 and T2 from the hexagon plots and the

Table 8. Method 5: Training Only the MLP vs. the Complete DNN

Test Data	T	Training Only the MLP						Training the Whole DNN					
		REC	PREC	SPEC	ACC	MCC	F1	REC	PREC	SPEC	ACC	MCC	F1
Medico Test Data	T1	0.9572	0.5859	0.9698	0.9692	0.7357	0.7269	0.9706	0.7773	0.9876	0.9868	0.8623	0.8633
	T2	0.9599	0.7804	0.9879	0.9867	0.8591	0.8609	0.9626	0.7809	0.9879	0.9868	0.8606	0.8623
	T3	0.9626	0.6316	0.9749	0.9744	0.7684	0.7627	0.9599	0.7771	0.9877	0.9865	0.8571	0.8589
CVC-12k	T1	0.6984	0.8972	0.5842	0.6799	0.2184	0.7854	0.8694	0.8507	0.2068	0.7625	0.0802	0.8599
	T2	0.7588	0.8993	0.5583	0.7265	0.2565	0.8231	0.9486	0.8571	0.1778	0.8242	0.1802	0.9005
	T3	0.7614	0.8933	0.5272	0.7236	0.2352	0.8221	0.9278	0.8462	0.1239	0.7981	0.0699	0.8851

T, the additional training dataset that was added to the Medico dataset; T1, Medico dataset + CVC-356-plus; T2, Medico dataset + CVC-612-plus; T3, Medico dataset + CVC-356-plus + CVC-612-plus.

corresponding ROC and PR curves. Although T2 shows a performance loss compared to T1 in Figure 10(i), Figure 10(j) shows that T2 achieves a performance improvement over T1. Next, we look for the reasons for these performance changes.

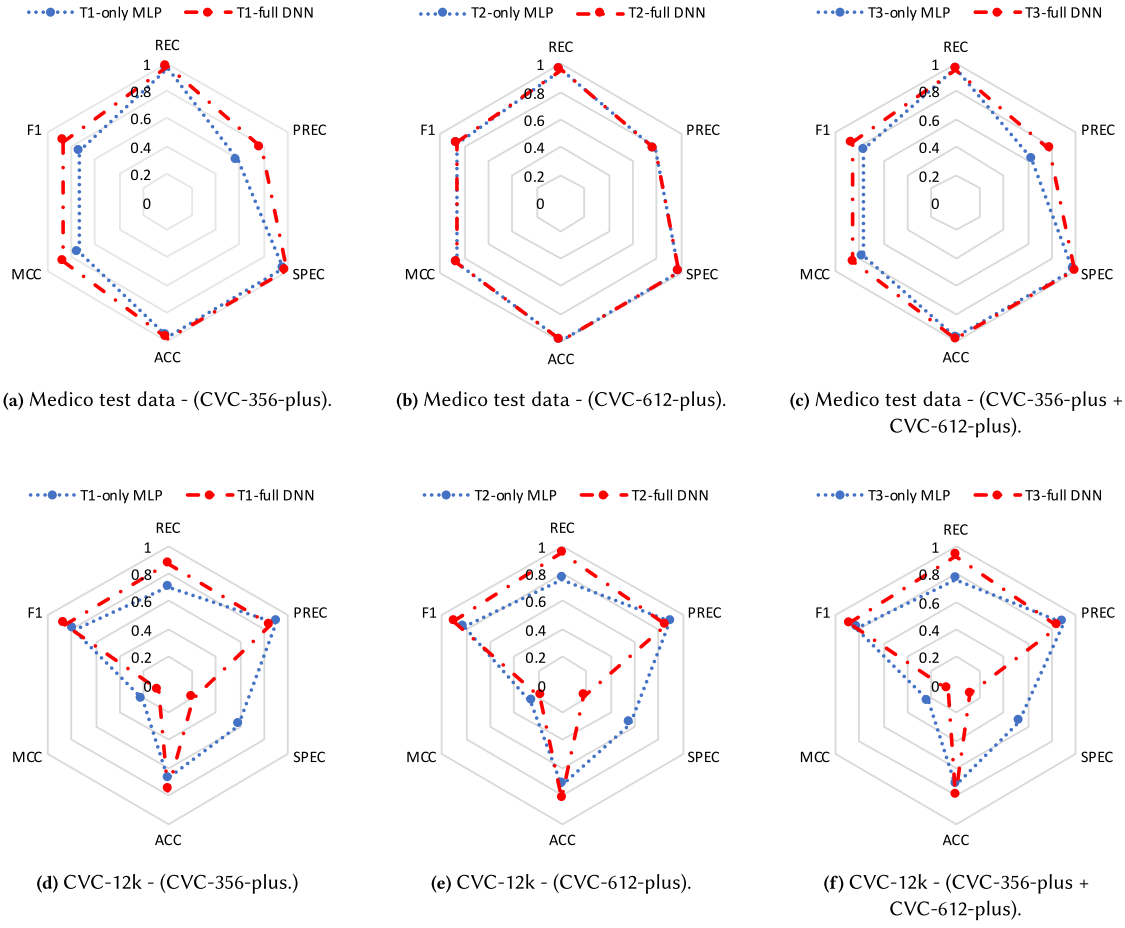
In method 5, the model with the 16 outputs corresponding to T1 has 15 choices to classify non-polyp images. Similarly, the Medico test dataset has more non-polyp images than polyp images. However, the model corresponding to T2 has a 50% chance to classify both polyps and non-polyps. As a result, the model of T1 shows better performance than the model of T2 in Figure 10(i). Because this shows a slight performance change, we cannot see the same difference in ROC and PR curves in Figure 11(a) and (c). In contrast, T2 in the plot in Figure 10(j) shows performance improvement when the model has a 50:50 chance for classifying polyps and non-polyps. This improvement occurred as a result of a large number of polyps in the CVC-12k dataset. The ROC and PR curves in plots in Figure 11(b) and (d) show this performance difference precisely. In other words, the model of T2 has a better chance of classifying polyps compared to the 1/16th chance in the model of T1.

The retrained models corresponding to T3, T4, and T5 do not show considerable performance changes for the Medico test dataset, as we can see from plots in Figure 10(i), (a), and (c). Conversely, the retraining method used in T3, T4, and T5 for the CVC-12k dataset shows large performance changes in the plots in Figure 10(j), (b), and (d). However, these methods show an overall performance loss. More comparisons on these plots are discussed in Section 6.

For the following experiments, we analyzed method 5 even further. The main focus of this analysis is to understand the behavior of the best model for training only the MLP versus training the whole DNN. In this experiment, we collected results for two main test datasets: the Medico test dataset and the CVC-12k dataset. Then, we collected performance measures from the two training mechanisms: training only the MLP and training the whole DNN. Furthermore, results were tabulated in Table 8, and corresponding graphs were depicted in Figure 12 to analyze them.

The first row of Figure 12 shows the differences in the performance of testing with the Medico test data. In the second row, it presents the performance changes for the CVC-12k dataset. The dotted lines in plots in Figure 12 represent training MLP. Similarly, the dashed lines represent training the whole DNN. The three plots of each row represent results of retraining the model with the Medico training data and CVC-356-plus dataset, the Medico training data and CVC-612-plus dataset, and the Medico training data and both CVC-356-plus and CVC-612-plus datasets, respectively.

According to the plots in Figure 12(a) through (c), it is clear that retraining the whole DNN can be used to improve the overall performance of the DNN model because we can see performance improvement in these plots except in Figure 12(b), which shows closely equal performance metrics. However, in test cases with the CVC-12k dataset, it shows a completely new behavior for retraining the whole DNN as depicted in Figure 12(d) through (f). These plots show large changes in the performance hexagons with considerable positive improvements for the



(a) Medico test data - (CVC-356-plus).

(b) Medico test data - (CVC-612-plus).

(c) Medico test data - (CVC-356-plus + CVC-612-plus).

(d) CVC-12k - (CVC-356-plus.)

(e) CVC-12k - (CVC-612-plus).

(f) CVC-12k - (CVC-356-plus + CVC-612-plus).

Fig. 12. Behavior of the complex DNN method (method 5) while training only MLP compared to training the whole DNN. The first row shows the effects for both cases when the test dataset is the Medico test data, and the second row shows the result when the test dataset is the CVC-12k dataset. T1, T2, and T3 represent the training dataset used for the model. (T1, Medico training dataset + CVC-356; T2, Medico training dataset + CVC-612; T3, Medico training dataset + CVC-356 + CVC-612.)

recall and considerable performance loss for the specificity values. This experiment also shows that researchers could be misled by the performance monitoring process of DNN methods using a single dataset. In other words, according to the first row of the figure, researchers may conclude that retraining the whole DNN is a positive factor. However, the results of the second row prove that it is not always true by showing performance losses for the same technique.

The results presented in plots in Figure 12 show difficulties in adapting ML models for cross-dataset generalization with a different perspective. In that experiment, the performance loss in specificity, which is a parameter of reflecting True Negative (TN) detection, shows that method 5 is affected by imbalanced data in the CVC-12k dataset. The main reason for the effect is that the CVC-12k dataset contains a lower percentage of negative images compared to positive ones. This reflects an important factor to take into account when developing generalizable ML models, which is that the ratio of negative and positive findings needs to be taken into account when looking at metrics. Metrics such as MCC are better suited to interpret results. In terms of ROC compared

to the PR curve, the results show that the PR curve reflects the performance of the model more realistically than ROC.

6 DISCUSSION

In this section, we present our findings and point out several important considerations for future research. Our discussion follows the same sequence as our contributions in this article.

In our experiments, combinations of ResNet-152, DenseNet-161, and an additional MLP produced the best result for the Medico 2018 dataset. The reported results from this model for the Medico Task led us to hold second place based on the MCC values calculated by organizers, and there was only a tiny gap of around 0.0029. Furthermore, the winning team of this competition used additional data items that were made by photo-editing tools for the imbalanced classes, such as the out-of-patients class. In contrast to this, our method 5 works well without using manually annotated data items because of the procedure we followed to implement and train that model. The procedure of implementing such a complex model is described step by step in Section 4.3, and anyone can follow these steps to get a well-performing DL model in a classification task.

In addition to the implementation and the procedure used in method 5, the data-filling mechanism used to fill the out-of-patient imbalance class shows impressive performance gain. This method is preferred when one class has a small number of data items in a multi-class classification task. In our work, without annotating more data ourself, which also requires the help of medical experts, we prefer to use random images from the Internet, as described in Section 5.1. This is an efficient way to add more data items without spending more time on manual annotation or creating synthetic data items. The preceding method works because the random images influence the ML models to make a wider range of possibilities to classify images into a particular class.

Dyed-lifted-polyps, dyed-resection-margins, esophagitis, and normal-z-line raised classification conflicts in our best method (method 5). If we could overcome these conflicts, then the model would perform better than the current recorded performance in the 2018 Medico Task. To identify the reasons for these classification conflicts, we manually investigated the images of these classes. If we compare sample images of dyed-lifted-polyps (Figure 1(c)), dyed-resection-margins (Figure 1(d)), esophagitis (Figure 1(e)), and normal-z-line (Figure 1(i)), then we can identify that this conflict was caused as a result of similar texture and shapes of these images. To overcome this problem, researchers can select only the images that made the conflict and train a new DL model to classify them into the correct classes. Then, this model can be added to the model introduced in method 5 using the property of its expandability.

Can we use our best DL model for real systems in hospitals to classify GI findings? Or can we use the state-of-the-art ML classification models introduced by researchers in real applications? Toward answering this question, this article focuses on deep evaluations of the proposed methods as one of the main contributions. Regularly, researchers present the performance of their classification models using only a test dataset, which was reserved from the dataset used to produce the training data. In addition, they measure the performance by selecting only a few measurements out of the REC, PREC, SPEC, ACC, MCC, and F1. However, we emphasize the requirements of an in-depth analysis of all of these six parameters at once to identify the real performance of ML models. Several of the works listed in Table 1 do not use this methodology as part of their evaluations. This makes it difficult to reason about the real-world performance of the proposed methods and how they compare with other methods. In this article, we also consider the importance of evaluating ML models with cross-datasets.

Why do we need cross-dataset evaluations? To explain this requirement, we consider the research work done by Wang et al. [75]. They presented an area under the ROC curve of 0.984 and a per-image sensitivity of 94.38 for polyp detection. In our first look, these results show a good DL model. Similarly, our results in Figure 10(i) and 11(a) and (c) reflect the same impression in the first look because it shows excellent performance as a DL model. However, after analyzing cross-dataset performance for polyp detection with a completely new dataset like CVC-12k, we recognized that performance gain is not enough for applying it in real applications. Therefore, from this work, we emphasize that researchers want to consider cross-dataset evaluations thoroughly before applying

their solutions in real-world applications. Otherwise, the selection bias, the capture bias, and the category bias (label bias) problems may appear in the results. Then, we may end up with the wrong conclusion about research works. All of these facts imply that more research must be performed to improve the generalizability along with the performance improvement on a single dataset or single data source.

7 CONCLUSION

We studied cross-dataset bias and evaluation metrics interpretation in ML using five methods and four different datasets within the field of GI endoscopy as respective use case. In particular, we performed an extensive study of ML models in the context of medical applications based on a use case of GI tract abnormality classification across different datasets. The main conclusion and resulting recommendation is that a multi-center or cross-dataset evaluation is important, if not essential, for ML models in the medical field to obtain a realistic understanding of the performance of such models in real-world settings.

We found that the combination of DNN ResNet-152 and DenseNet-161 with an additional MLP performed best on both the validation and test datasets. This model shows that a combination of multiple pre-trained DNN models can have better capabilities to classify images into the correct classes because of their cumulative decision-making capabilities. We also proposed an evaluation method using six measures: REC, PREC, SPEC, ACC, MCC, and F1. Moreover, we suggest that these measures should be presented all at once using hexagon plots that convey a complete view of real performance. It is our hope that these tools can enable a more realistic evaluation and comparison of ML methods.

Furthermore, we presented cross-dataset evaluations to identify the generalizability of our ML models, emphasizing the fact that achieving high scores for evaluation metrics does not always represent the real performance of ML models and should be interpreted with care. By evaluating the ML models with cross-datasets experiments, we showed the complexity of understanding the real functional performance of the models. The state-of-the-art research works that perform classification cannot be used in practical applications because of their lack of generalizability. Based on the experimental results, we conclude that researchers should focus on implementing and researching generalizable ML models with cross-dataset evaluations. Rather than presenting metrics calculated from a simple training and testing split of the data, we suggest to always rely on cross-dataset evaluation to obtain a real-world representative indication of model performance. This is especially important in a medical context because one has to make sure that the obtained models are reliable and not just perform well on a specific dataset.

Finally, we want to point out that the lack of generalization, as evidenced by the poor result for cross-dataset evaluation presented in this article, rises a very important question: in the context of cross-dataset or multi-center studies, is it really possible to have generalizable ML models? This is something that we ourselves plan to investigate further in future work, and it is ou hope that other researchers in computer science and medicine will do the same or at least have the question in their mind when performing similar studies.

ACKNOWLEDGMENTS

We would like to thank the reviewers for their contributions to the article.

REFERENCES

[1] Taruna Agrawal, Rahul Gupta, Saurabh Sahu, and Carol Y. Espy-Wilson. 2017. SCL-UMD at the Medico Task-MediaEval 2017: Transfer learning based classification of medical images. In *Proceedings of MediaEval 2017*.

[2] Luís A. Alexandre, Nuno Nobre, and João Casteleiro. 2008. Color and position versus texture features for endoscopic polyp detection. In *Proceedings of IEEE BMEI2008*, Vol. 2. 38–42.

[3] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. 2009. Texture-based polyp detection in colonoscopy. In *Bildverarbeitung für die Medizin 2009*. 346–350.

[4] Quentin Angermann, Jorge Bernal, Cristina Sánchez-Montes, Maroua Hammami, Gloria Fernández-Esparrach, Xavier Dray, Olivier Romain, F. Javier Sánchez, and Aymeric Histace. 2017. Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis. In *Proceedings of CARE and CLIP 2017*. 29–41.

[5] Jorge Bernal, Aymeric Histace, Marc Masana, Quentin Angermann, Cristina Sánchez-Montes, Cristina Rodriguez, Maroua Hammami, et al. 2018. Polyp detection benchmark in colonoscopy videos using GTCreator: A novel fully configurable tool for easy and fast annotation of image databases. In *Proceedings of CARS 2018*.

[6] Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. 2015. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics* 43 (2015), 99–111.

[7] Jorge Bernal, Javier Sánchez, and Fernando Vilarino. 2012. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognition* 45, 9 (2012), 3166–3182.

[8] Jorge Bernal, Javier Sánchez, and Fernando Vilarino. 2013. Impact of image preprocessing methods on polyp localization in colonoscopy frames. In *Proceedings of IEEE EMBC 2013*. 7350–7354.

[9] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sánchez, Bogdan J. Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, et al. 2017. Comparative validation of polyp detection methods in video colonoscopy: Results from the MICCAI 2015 Endoscopic Vision Challenge. *IEEE Transactions on Medical Imaging* 36, 6 (2017), 1231–1249.

[10] Rune Johan Borgli, Pål Halvorsen, Michael Riegler, and Håkon Kvale Stensland. 2018. Automatic hyperparameter optimization in Keras for the MediaEval 2018 Medico Multimedia Task. In *Proceedings of MediaEval 2018*.

[11] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*. 177–186.

[12] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L. Siegel, Lindsey A. Torre, and Ahmedin Jemal. 2018. 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 (2018), 394–424.

[13] Da-Chuan Cheng, Wen-Chien Ting, Yung-Fu Chen, and Xiaoyi Jiang. 2011. Automatic detection of colorectal polyps in static images. *Biomedical Engineering: Applications, Basis and Communications* 23, 05 (2011), 357–367.

[14] Torch Contributors. 2018. Torchvision Models. Retrieved May 7, 2020 from https://pytorch.org/docs/stable/torchvision/models.html.

[15] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of Operations Research* 134 (2005), 19–67.

[16] Thomas de Lange, Pål Halvorsen, and Michael Riegler. 2018. Methodology to develop machine learning algorithms to improve performance in gastrointestinal endoscopy. *World Journal of Gastroenterology* 24, 45 (2018), 5057.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE CVPR 2009*. 248–255.

[18] Danielle Dias and Ulisses Dias. 2018. Transfer learning with CNN architectures for classifying gastrointestinal diseases and anatomical landmarks. In *Proceedings of MediaEval 2018*.

[19] Patrick Doetsch, Christian Buck, Pavlo Golik, Niklas Hoppe, Michael Kramp, Johannes Laudenberg, Christian Oberdörfer, Pascal Steingrube, Jens Forster, and Arne Mauser. 2009. Logistic model trees with AUC split criterion for the KDD Cup 2009 Small Challenge. In *Proceedings of KDD-Cup '09*. 77–88.

[20] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian detection: A benchmark. In *Proceedings of IEEE CVPR 2009*.

[21] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2000. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics* 28, 2 (2000), 337–407.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of IEEE CVPR 2016*. 770–778.

[24] Steven A. Hicks, Pia H. Smedsrud, Pål Halvorsen, and Michael Riegler. 2018. Deep learning based disease detection using domain specific transfer learning. In *Proceedings of MediaEval 2018*.

[25] Trung-Hieu Hoang, Hai-Dang Nguyen, and Thanh-An Nguyen. 2018. An application of residual network and faster - RCNN for Medico: Multimedia Task at MediaEval 2018. In *Proceedings of MediaEval 2018*.

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of IEEE CVPR 2017*. 2261–2269.

[27] Sae Hwang, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C. De Groen. 2007. Polyp detection in colonoscopy video using elliptical shape feature. In *Proceedings of IEEE ICIP 2007*, Vol. 2. 465–468.

[28] Dimitrios K. Iakovidis, Dimitrios E. Maroulis, Stavros A. Karkanis, and A. Brokos. 2005. A comparative study of texture features for the discrimination of gastric polyps in endoscopic video. In *Proceedings of IEEE CBMS 2005*. 575–580.

[29] Yuji Iwahori, Takayuki Shinohara, Akira Hattori, Robert J. Woodham, Shinji Fukui, Manas Kamal Bhuyan, and Kunio Kasugai. 2013. Automatic polyp detection in endoscope images using a Hessian filter. In *Proceedings of MVA 2013*, Vol. 13. 21–24.

[30] Debesh Jha, Pia Smedsrud, Michael Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard Johansen. 2020. Kvasir-SEG: A segmented polyp dataset. In *Proceedings of MMM 2020*. 1–12.

[31] Xiao Jia and Max Q.-H. Meng. 2017. Gastrointestinal bleeding detection in wireless capsule endoscopy images using handcrafted and CNN features. In *Proceedings of IEEE EMBC 2017*. 3154–3157.

[32] Stavros A. Karkanis, Dimitrios K. Iakovidis, Dimitrios E. Maroulis, Dimitris A. Karras, and M. Tzivras. 2003. Computer-aided tumor detection in endoscopic video using color wavelet features. *IEEE Transactions on Information Technology in Biomedicine* 7, 3 (2003), 141–152.

[33] Zeshan Khan and Muhammad Atif Tahir. 2018. Majority voting of heterogeneous classifiers for finding abnormalities in the gastro-intestinal tract. In *Proceedings of MediaEval 2018.*

[34] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei Efros, and Antonio Torralba. 2012. Undoing the damage of dataset bias. In *Proceedings of ECCV 2012.*

[35] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

[36] Mathias Kirkerød, Vajira Thambawita, Michael Riegler, and Pål Halvorsen. 2018. Using preprocessing as a tool in medical image detection. In *Proceedings of MediaEval 2018.*

[37] Tobey H. Ko, Zhonglei Gu, and Yang Liu. 2018. Weighted discriminant embedding: Discriminant subspace learning for imbalanced medical data classification. In *Proceedings of MediaEval 2018.*

[38] Niels Landwehr, Mark Hall, and Eibe Frank. 2005. Logistic model trees. *Machine Learning* 59, 1–2 (2005), 161–205.

[39] A. M. Leufkens, M. G. H. van Oijen, F. P. Vleggaar, and P. D. Siersema. 2012. Factors influencing the miss rate of polyps in a back-to-back colonoscopy study. *Endoscopy* 44, 05 (2012), 470–475.

[40] Baopu Li and Max Q.-H. Meng. 2012. Tumor recognition in wireless capsule endoscopy images using textural features and SVM-based feature selection. *IEEE Transactions on Information Technology in Biomedicine* 16, 3 (2012), 323–329.

[41] David Lieberman. 2005. Quality and colonoscopy: A new imperative. *Gastrointestinal Endoscopy* 61, 3 (2005), 392–394.

[42] Mathias Lux, Michael Riegler, Pål Halvorsen, Konstantin Pogorelov, and Nektarios Anagnostopoulos. 2016. LIRE: Open source visual information retrieval. In *Proceedings of ACM MMSys 2016.* 30.

[43] Alexander V. Mamonov, Isabel N. Figueiredo, Pedro N. Figueiredo, and Yen-Hsi Richard Tsai. 2014. Automated polyp detection in colon capsule endoscopy. *IEEE Transactions on Medical Imaging* 33, 7 (2014), 1488–1502.

[44] Yuichi Mori and Shin-Ei Kudo. 2018. Detecting colorectal polyps via machine learning. *Nature Biomedical Engineering* 2, 10 (2018), 713.

[45] Syed Sadiq Ali Naqvi, Shees Nadeem, Muhammad Zaid, and Muhammad Atif Tahir. 2017. Ensemble of texture features for finding abnormalities in the gastro-intestinal tract. In *Proceedings of MediaEval 2017.*

[46] Olga Ostroukhova, Konstantin Pogorelov, Michael Riegler, Duc-Tien Dang-Nguyen, and Pål Halvorsen. 2018. Transfer learning with prioritized classification and training dataset equalization for medical objects detection. In *Proceedings of MediaEval 2018.*

[47] Sun Young Park, Dustin Sargent, Inbar Spofford, Kirby G. Vosburgh, and Y. A-Rahim. 2012. A colon video analysis framework for polyp detection. *IEEE Transactions on Biomedical Engineering* 59, 5 (2012), 1408.

[48] Stefan Petscharnig, Klaus Schöffmann, and Mathias Lux. 2017. An inception-like CNN architecture for GI disease and anatomical landmark classification. In *Proceedings of MediaEval 2017.*

[49] Konstantin Pogorelov, Olga Ostroukhova, Mattis Jeppsson, Håvard Espeland, Carsten Griwodz, Thomas de Lange, Dag Johansen, Michael Riegler, and Pål Halvorsen. 2018. Deep learning and hand-crafted feature based approaches for polyp detection in medical videos. In *Proceedings of IEEE CBMS 2018.* 381–386.

[50] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Carsten Griwodz, Dag Johansen, Concetto Spampinato, et al. 2017. Nerthus: A bowel preparation quality video dataset. In *Proceedings of ACM MMSys 2017.* 170–174.

[51] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, et al. 2017. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of ACM MMSys 2017.* 164–169.

[52] Konstantin Pogorelov, Michael Riegler, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Carsten Griwodz, Peter Thelin Schmidt, and Pål Halvorsen. 2017. Efficient disease detection in gastrointestinal videos—Global features versus neural networks. *International Journal of Multimedia Tools and Applications* 76, 21 (2017), 22493–22525.

[53] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Thomas De Lange, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, and Olga Ostroukhova. 2018. Medico multimedia task at MediaEval 2018. In *Proceedings of MediaEval 2018.*

[54] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Carsten Griwodz, Thomas de Lange, Kristin Randel, Sigrun Eskeland, Dang Nguyen, Duc Tien, and Olga Ostroukhova. 2017. A comparison of deep learning with global features for gastrointestinal disease detection. In *Proceedings of MediaEval 2017.*

[55] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Thomas De Lange, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, and Olga Ostroukhova. 2018. Medico multimedia task at MediaEval 2018. In *Proceedings of MediaEval 2018.*

[56] Michael Riegler, Martha Larson, Mathias Lux, and Christoph Kofler. 2014. How 'how' reflects what's what: Content-based exploitation of how users frame social images. In *Proceedings of ACM MM 2014.* 397–406.

[57] Michael Riegler, Mathias Lux, Carsten Griwodz, Concetto Spampinato, Thomas de Lange, Sigrun L. Eskeland, Konstantin Pogorelov, et al. 2016. Multimedia and medicine: Teammates for better disease detection and survival. In *Proceedings of ACM MM 2016.* 968–977.

[58] Michael Riegler, Konstantin Pogorelov, Sigrun Losada Eskeland, Peter Thelin Schmidt, Zeno Albisser, Dag Johansen, Carsten Griwodz, Pål Halvorsen, and Thomas De Lange. 2017. From annotation to computer-aided diagnosis: Detailed evaluation of a medical multimedia system. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 3 (2017), 26.

[59] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Kristin Randel, Sigrun Losada Eskeland, Duc-Tien Dang-Nguyen, Mathias Lux, Carsten Griwodz, Concetto Spampinato, and Thomas Lange. 2017. Multimedia for medicine: The Medico Task at MediaEval 2017. In *Proceedings of MediaEval 2017*.
[60] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747.
[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (2015), 211–252.
[62] Steven L. Salzberg. 1994. C4. 5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* 16, 3 (1994), 235–240.
[63] Younghak Shin and Ilangko Balasingham. 2017. Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. In *Proceedings of IEEE EMBC 2017*. 3277–3280.
[64] Michael Steiner, Mathias Lux, and Pål Halvorsen. 2018. The 2018 Medico Multimedia Task submission of Team NOAT using neural network features and search-based classification. In *Proceedings of MediaEval 2018*.
[65] Marc Sumner, Eibe Frank, and Mark Hall. 2005. Speeding up logistic model tree induction. In *Proceedings of PKDD 2005*. 675–683.
[66] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of ICML 2013*. 1139–1147.
[67] Nima Tajbakhsh, Suryakanth R. Gurudu, and Jianming Liang. 2015. Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks. In *Proceedings of IEEE ISBI 2015*. 79–83.
[68] Nima Tajbakhsh, Suryakanth R. Gurudu, and Jianming Liang. 2016. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging* 35, 2 (2016), 630–644.
[69] Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging* 35, 5 (2016), 1299–1312.
[70] Mario Taschwer, Manfred Jürgen Primus, Klaus Schoeffmann, and Oge Marques. 2018. Early and late fusion of classifiers for the MediaEval Medico Task. In *Proceedings of MediaEval 2018*.
[71] Vajira Thambawita, Debesh Jha, Michael Riegler, Pål Halvorsen, Hugo Lewi Hammer, Håvard D. Johansen, and Dag Johansen. 2018. The Medico-Task 2018: Disease detection in the gastrointestinal tract using global features and deep learning. In *Proceedings of MediaEval 2018*.
[72] A. Torralba and A. A. Efros. 2011. Unbiased look at dataset bias. In *Proceedings of IEEE CVPR 2011*. 1521–1528.
[73] David Vázquez, Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Antonio M. López, Adriana Romero, Michal Drozdzal, and Aaron C. Courville. 2017. A benchmark for endoluminal scene segmentation of colonoscopy images. *Journal of Healthcare Engineering* 2017 (2017), 4037190.
[74] L. Von Karsa, J. Patnick, and N. Segnan. 2012. European guidelines for quality assurance in colorectal cancer screening and diagnosis. First edition—Executive summary. *Endoscopy* 44, Suppl. 3 (2012), SE1–SE8.
[75] Pu Wang, Xiao Xiao, Jeremy R. Glissen Brown, Tyler M. Berzin, Mengtian Tu, Fei Xiong, Xiao Hu, et al. 2018. Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. *Nature Biomedical Engineering* 2, 10 (2018), 741.
[76] Yi Wang, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C. De Groen. 2014. Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy. *IEEE Journal of Biomedical and Health Informatics* 18, 4 (2014), 1379–1389.
[77] Yi Wang, Wallapak Tavanapong, Johnny Wong, Jung Hwan Oh, and Piet C. De Groen. 2015. Polyp-Alert: Near real-time feedback during colonoscopy. *International Journal of Computer Methods and Programs in Biomedicine* 120, 3 (2015), 164–179.
[78] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng Ann Heng. 2017. Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos. *IEEE Journal of Biomedical and Health Informatics* 21, 1 (2017), 65–75.
[79] Yixuan Yuan, Dengwang Li, and Max Q.-H. Meng. 2018. Automatic polyp detection via a novel unified bottom-up and top-down saliency approach. *IEEE Journal of Biomedical and Health Informatics* 22, 4 (2018), 1250–1260.
[80] Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. arXiv:1212.5701.
[81] Xu Zhang, Fei Chen, Tao Yu, Jiye An, Zhengxing Huang, Jiquan Liu, Weiling Hu, Liangjing Wang, Huilong Duan, and Jianmin Si. 2019. Real-time gastric polyp detection using convolutional neural networks. *PloS One* 14, 3 (2019), e0214133.
[82] Xu Zhang, Weiling Hu, Fei Chen, Jiquan Liu, Yuanhang Yang, Liangjing Wang, Huilong Duan, and Jianmin Si. 2017. Gastric precancerous diseases classification using CNN with a concise model. *PloS One* 12, 9 (2017), e0185508.
[83] Mingda Zhou, Guanqun Bao, Yishuang Geng, Bader Alkandari, and Xiaoxi Li. 2014. Polyp detection and radius measurement in small intestine using video capsule endoscopy. In *Proceedings of IEEE BMEI2014*. 237–241.

Received March 2019; revised December 2019; accepted February 2020