Automatic Disease Detection and Report Generation for Gastrointestinal Tract Examinations

Philipp Harzig, Moritz Einfalt, Rainer Lienhart
[philipp.harzig,moritz.einfalt,rainer.lienhart]@informatik.uni-augsburg.de
University of Augsburg
Augsburg, Germany

ABSTRACT

In this paper, we present a method to automatically identify diseases from videos of gastrointestinal (GI) tract examinations using a Deep Convolutional Neural Network (DCNN) that processes images from digital endoscopes. Our goal is to aid domain experts by automatically detecting abnormalities and generating a report that summarizes the main findings. We have implemented a model that uses two different DCNN architectures to generate our predictions, which are also capable of running on a mobile device [1]. Using this architecture, we are able to predict findings on individual images. Combined with class activations maps (CAM), we can also automatically generate a textual report describing a video in detail while giving hints about the spatial location of findings and anatomical landmarks. Our work shows one way to use a multi-disease detection pipeline to also generate video reports that summarize key findings.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; **Video summarization**; **Supervised learning by classification**; *Neural networks.*

KEYWORDS

gastrointestinal tract, GI, video summarization, medical disease detection, textual medical report generation

ACM Reference Format:
Philipp Harzig, Moritz Einfalt, Rainer Lienhart. 2019. Automatic Disease Detection and Report Generation for Gastrointestinal Tract Examinations. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3343031.3356066

1 INTRODUCTION

In this work, we present our method for automatic identification of diseases and anatomical landmarks in the human digestive system. Gastroscopy and colonoscopy are real-time examinations of the human gastrointestinal (GI) tract by using digital endoscopes. Analyzing these videos is both time-consuming and needs a domain expert. Supporting these domain experts by machine learning techniques is one promising approach to reduce costs.

This work gives a technical overview of our submission to the 2019 ACM Multimedia Grand Challenge BioMedia: Multimedia in Medicine [4]. This challenge focuses on automatically detecting normal findings, abnormalities and anatomical landmarks in GI tract images composed from two different datasets. Pogorelov et al. [10] presented a multi-class dataset consisting of GI tract images, which has greater variability than other publicly available datasets [1, 12]. In addition, their dataset contains other classes than only focusing on polyps, i.e., classes related to polyp removal and three anatomical landmarks of the GI tract. Pogorelov et al. [9] also presented a dataset consisting of classes that allow to assess whether the bowel was cleansed sufficiently before colonoscopy.

2 DATASET

For our work, we use the Medico 2018 dataset provided by the challenge. The dataset includes images for 16 classes and consists of a development split and a test split with 5,293 and 8,740 images, respectively. The classes represent anatomical landmarks, pathological findings or endoscopic procedures in the GI tract. The Medico 2018 dataset comprises parts of the Kvasir [10] and the Nerthus [9] dataset. The different classes are quite balanced with the exception of the *out-of-patient* and *instruments* class, which only account for 0.076% (4 images) and 0.680% (36 images) of the development split, respectively. For our models, we create a train and validation split from the development set with a ratio of $3:1$, respectively.

In addition, we use the Kvasir-v2 dataset [10] for two of our models to improve detection accuracy. The Kvasir-v2 dataset provides 8000 additional images covering 8 classes. For the report generation subtask an additional dataset consisting of six videos has been provided by the BioMedia challenge organizers.

3 METHOD

We use two different CNNs to extract features from the input images, i.e., the MobileNetV2 [11] and the DenseNet-121 [5]. In order to detect the class of the input image, we append two fully-connected layers to the average-pooled feature map $\mathbf{f} \in \mathbb{R}^k$ of the CNN with k being the depth of the respective CNN's feature map. The development dataset is labeled as a single-classification problem, i.e., one correct class per image. Thus, we train the first fully-connected layer fc_1 with a Softmax (σ) cross-entropy loss function

$$L_{\text{softmax}} = -\sum_{x} \text{gt}(x) \log \sigma(\text{fc}_1(\mathbf{f})) \tag{1}$$

[1] We ported one of our MobileNetV2 models to be usable on an iPhone.

with gt(x) being the ground-truth label for training sample x. Nevertheless, a single image could still have multiple correct classifications, which we try to model with a second prediction module. For instance, an image could depict a pathological finding and an instrument. We train the second fully-connected layer with a Sigmoid cross-entropy loss, which allows us to output the likelihood of every class instead of predicting the most probable class only. However, predicting probabilities for independent classes does not impose a ranking on those classes. Hence, we always use the prediction of the fully-connected layer trained with the Softmax cross-entropy loss first. We select the class-specific thresholds for the predictions trained with the Sigmoid cross-entropy loss by using the thresholds that yield the highest F1-score. If there are additional predictions after applying the class-specific thresholds, we output those according to the Softmax ranking.

3.1 Training

While training, we resize every image to a size of 256×256 and extract a random crop of size 224×224. Additionally, for every image, we randomly decide whether to rotate the image by 90°, to flip it horizontally and to flip it vertically, which results in 8 possible configurations for every input image.

We employ a two-stage training. In the first stage, we freeze the weights of the feature extractor CNN and only train the two fully-connected layers. We use the Adam [7] optimizer with a learning rate $\eta = 0.001$ and train for up to 100 epochs with early stopping. In the second stage, we unfreeze the weights of the CNN and train with $\eta = 0.0001$ and an exponential learning rate schedule, which decays the learning rate by 0.5 every 20 epochs. In the finetuning stage, we also use an L2 regularization loss with a multiplier of $1 \cdot 10^{-4}$ for all convolution and dense layer weights. We select our final model with an early stopping strategy, i.e., we choose the model with the best accuracy on the validation set.

4 REPORT GENERATION

Medical report generation is a task with which is dealt in other areas as well. For instance, with the release of the Indiana University X-ray dataset [2], many works deal with connecting natural language and chest X-ray images. For instance, Jing et al. [6] and Harzig et al. [3] use a hierarchical Long Short-Term Memory (HLSTM) [8] model, which allows to generate multiple sentences to form a paragraph that reflects a doctor's report. However, in contrast to our problem, this dataset [2] contains chest X-ray images combined with natural language reports of doctors. We, however, cannot use a language model like a Recurrent Neural Network (RNN), trained on natural language paragraphs.

The provided dataset contains six videos which span from 00:01–05:11 (39–7,783 frames) from which we extract each frame with the FFmpeg library. Then, we use our trained model to predict the class for each frame. We smooth the predictions over 30 frames in the past and future using a simple algorithm: Given the future and past frames, we determine the most probable prediction by removing outliers using the z-score[2]. Whenever the prediction of the current frame is the same as the most probable future prediction and different from the most probable past prediction, we change the classification result of our smoothed prediction.

For each continuous sequence of frames for which the same classification result was predicted, we create a so-called video section. For example, we identified 19 consecutive video sections in Figure 3.

In addition, we use class activation maps (CAM) [13] to localize class-specific image regions, i.e., we infer the regions that contributed most to the classification outcome. The CAM can be seen as a probability distribution over the CNN's output feature map $\mathbf{f}$. We average these probabilities over each video section that we identified in the video. By using these averaged probability distributions, we then identify the area (one of top-left, top-right, bottom-left, bottom-right or center) that seems to be mostly responsible for the classification. We visualized this process in Figure 1.

Our final report for each video consists of three sections, (1) the main findings, (2) a brief summary and (3) a detailed summary. In the main findings, we provide the two most probable classifications over the whole video sequence together with their respective frequency of occurrence. Second, we provide a brief summary which explains all consecutive classifications of video sections in a chronological order. Finally, we give a detailed summary that describes every event within the video sequence with an exact time span, the classification result and the spatial location in which the event has been detected with the highest probability.

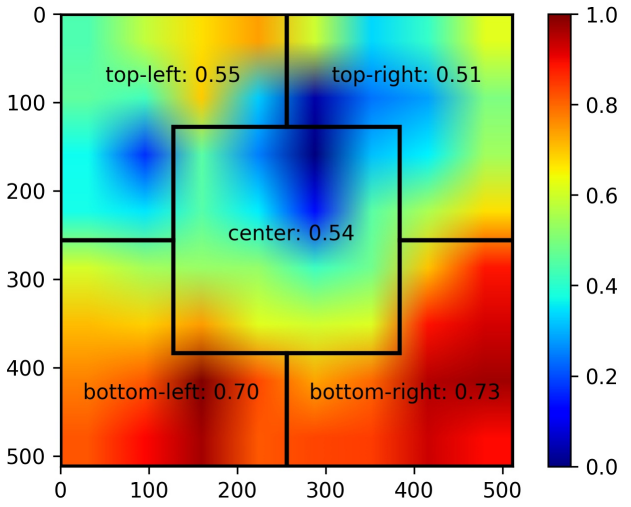

Figure 1: Average class activation map (CAM) for a video segment together with an overlay describing our five areas of interest. The area with the highest attention score is chosen as the most interesting area within the current video segment. Note that all region scores are normalized according to their respective areas.

5 RESULTS

For our submitted models, we use two different training datasets. For the *-ver1* models, we use 75% (3969 images) of the provided Medico development dataset for training while keeping 25% for validating our model and selecting the best performing one. We

[2]The z-score of a random variable $\mathbf{X}$ is defined as $Z = \frac{\mathbf{X} - E[\mathbf{X}]}{\sigma(\mathbf{X})}$.

Table 1: Results of all our models on the detection (*detection-ver?*) and efficient detection (*speed-ver?*) subtasks. *detection-ver2* is our model with the best performance according to the MCC score. We report the individual times for the efficient detection subtasks in Table 2.

model	TP	TN	FP	FN	precision	recall	specificity	F1	MCC
detection-ver1	8291	130609	446	446	**0.94442**	**0.90053**	0.99664	**0.91054**	0.94332
detection-ver2	**8419**	**130737**	**318**	**318**	0.89897	0.88458	**0.99752**	0.88471	**0.95974**
speed-ver1	8108	130426	629	629	0.86063	0.85300	0.99514	0.85142	0.92009
speed-ver2	8375	130693	362	362	0.89534	0.88129	0.99713	0.87993	0.95429
hardware	8108	130426	629	629	0.86063	0.85300	0.99514	0.85142	0.92009

Table 2: Official timings for our models. All times t are measured in milliseconds (ms). fps stands for frames per second.

model	t_{avg}	t_{min}	t_{max}	fps_{avg}	fps_{min}	fps_{max}
speed-ver1	**0.3087**	0.1219	18.1812	**3238.87**	55.00	8204.27
speed-ver2	0.3100	0.1342	17.3601	3226.04	57.60	7453.23
hardware	0.7862	**0.1090**	**9.3824**	1271.98	**106.58**	**9175.40**

extend the training split by the Kvasir-v2 [10] dataset (3969 + 8000 = 11969 images) for our *-ver2* models.

5.1 Detection Subtask

We submitted two models for the detection subtask, namely *detection-ver1* and *detection-ver2*. We trained *detection-ver1* with a MobileNetV2 with a width multiplier of 1.4. Even though the MobileNetV2 is designed as a mobile architecture, we found it to perform better than a DenseNet-121 and or a DenseNet-201 when only using the train split of the Medico development dataset. For *detection-ver2*, we used the DenseNet-121 CNN that achieved better results on the validation split. We depict our results in Table 1 and see that *detection-ver2* performs better for almost every metric except precision, recall and the F1-score. As we can see in Table 3 this is caused by the underrepresented *out-of-patient* class, which does not get detected by the *detection-ver2* model.

Our models are able to output multiple detections, e.g., there might be cases where a finding and an instrument is detected. However, detections for multiple classes is constrained as there are no multi-class annotations as of now.

5.2 Efficient Detection Subtask

Similar to the detection subtask, we submitted two models for the efficient detection subtask, which make use of the two different dataset variants, which we proposed in Section 5. We use a MobileNetV2 with a width multiplier of 1.0 for our efficient detection models, which allows for faster detection times while sacrificing a bit of accuracy. However, the Matthews correlation coefficient (MCC) score for the model *speed-ver2* is almost on par with the *detection-ver2* model. In contrast, when using the smaller dataset, i.e., only the train split of the Medico development dataset, the performance decreases by over two percent when using the MobileNetV2 with the smaller width multiplier. In Table 2, we list the times our models take to classify a single image. *speed-ver1* and *speed-ver2* are our submitted models with an average detection time for a single image of 0.3087 ms and 0.3100 ms, respectively. We measured those times on a dual-CPU workstation with 48 threads and a single NVIDIA TITAN X (Pascal) GPU. The *hardware* model is the same as *speed-ver1*, but was submitted to the organizers of the challenge as a docker image to be comparable with other submissions in terms of hardware configuration. In this scenario, the average processing time per image takes longer with 0.7862 ms, but the minimal processing time for an image is shorter with 0.1090 ms compared to 0.1219 ms for model *speed-ver1*.

Main findings:
==========
The video mostly shows esophagitis (70.85%), followed by blurry-nothing (14.88%).
Brief summary:
==========
The video sequence shows the following events in this chronological order: colon-clear, blurry-nothing, esophagitis, normal-z-line, esophagitis.
Detailed summary:
==========

FROM - TO	Description of current time period within the video.
00:00-00:00	An inflammation of the esophagus is visible mostly in the center (Esophagitis).
00:00-00:01	A clear colon can be seen mostly in the center.
00:01-00:02	An inflammation of the esophagus is visible mostly in the center (Esophagitis).
00:02-00:04	A clear colon can be seen mostly in the center.
00:04-00:09	The image is blurry and it is hard to identify what currently can be seen.
00:09-00:09	Instruments are visible within the current section of the video mostly in the bottom-left.
00:09-00:09	The image is blurry and it is hard to identify what currently can be seen.
00:09-00:10	retroflex-rectum mostly in the top-left.
00:10-00:11	The image is blurry and it is hard to identify what currently can be seen.
00:11-00:15	An inflammation of the esophagus is visible mostly in the center (Esophagitis).
00:15-00:15	A normal z-line can be seen mostly in the top-right.
00:15-00:31	An inflammation of the esophagus is visible mostly in the center (Esophagitis).
00:31-00:32	The image is blurry and it is hard to identify what currently can be seen.
00:32-00:44	An inflammation of the esophagus is visible mostly in the center (Esophagitis).
00:44-00:46	A normal z-line can be seen mostly in the center.
00:46-00:47	An inflammation of the esophagus is visible mostly in the top-right (Esophagitis).
00:47-00:48	Dyed resesected margins can be seen mostly in the top-left.
00:48-00:49	Instruments are visible within the current section of the video mostly in the bottom-left.
00:49-00:51	An inflammation of the esophagus is visible mostly in the center (Esophagitis).

Figure 2: Generated report for 3e3a7ac0-4244-46cc-89a1-44ce84dd1ccf.avi. This report matches with the smoothed prediction (bottom bar) of Figure 3.

5.3 Report Generation

For generating reports, we used the *detection-ver1* model from Section 5.1. As we already described in Section 4, we extracted all frames for each given video and predicted their most probable class label. We depict such a classification result for one video in Figure 3, where the top bar shows the raw classifications for every frame. The bottom bar shows the predictions which were smoothed over a window of 30 frames in the future and past. In the Figure, we also depict one image representative for each extracted video section. Together with a text template library and identifying the region

Table 3: Main metrics listed by class. We report the results of models *detection-ver1* and *detection-ver2* seperated by /.

class	TP	TN	FP	FN	precision	recall	specificity	F1
blurry-nothing	37/35	8698/8698	0/2	2/2	0.94872/0.94595	1.00000/0.94595	1.00000/0.99977	0.97368/0.94595
colon-clear	1065/1065	7660/7634	0/0	12/38	0.98886/0.96555	1.00000/1.00000	1.00000/1.00000	0.99440/0.98247
dyed-lifted-polyps	520/540	8101/8130	36/16	80/51	0.86667/0.91371	0.93525/0.97122	0.99558/0.99804	0.89965/0.94159
dyed-resection-margins	535/564	8122/8142	29/0	51/31	0.91297/0.94790	0.94858/1.00000	0.99644/1.00000	0.93043/0.97325
esophagitis	462/543	8132/8180	94/13	49/1	0.90411/0.99816	0.83094/0.97662	0.98857/0.99841	0.86598/0.98727
instruments	131/125	8464/8464	142/148	0/0	1.00000/1.00000	0.47985/0.45788	0.98350/0.98281	0.64851/0.62814
normal-cecum	570/582	8136/8149	14/2	17/4	0.97104/0.99317	0.97603/0.99658	0.99828/0.99975	0.97353/0.99487
normal-pylorus	560/561	8171/8176	1/0	5/0	0.99115/1.00000	0.99822/1.00000	0.99988/1.00000	0.99467/1.00000
normal-z-line	512/562	8082/8162	51/1	92/12	0.84768/0.97909	0.90941/0.99822	0.99373/0.99988	0.87746/0.98857
out-of-patient	1/0	8735/8735	1/2	0/0	1.00000/0.00000	0.50000/0.00000	0.99989/0.99977	0.66667/0.00000
polyps	365/373	8261/8295	9/1	102/68	0.78158/0.84580	0.97594/0.99733	0.99891/0.99988	0.86801/0.91534
retroflex-rectum	184/179	8535/8544	8/13	10/1	0.94845/0.99444	0.95833/0.93229	0.99906/0.99848	0.95337/0.96237
retroflex-stomach	394/394	8338/8336	3/3	2/4	0.99495/0.98995	0.99244/0.99244	0.99964/0.99964	0.99369/0.99119
stool-inclusions	494/468	8221/8181	12/38	10/50	0.98016/0.90347	0.97628/0.92490	0.99854/0.99538	0.97822/0.91406
stool-plenty	1956/1886	6771/6772	9/79	1/0	0.99949/1.00000	0.99542/0.95980	0.99867/0.98847	0.99745/0.97949
ulcerative-colitis	505/542	8182/8139	37/0	13/56	0.97490/0.90635	0.93173/1.00000	0.99550/1.00000	0.95283/0.95088

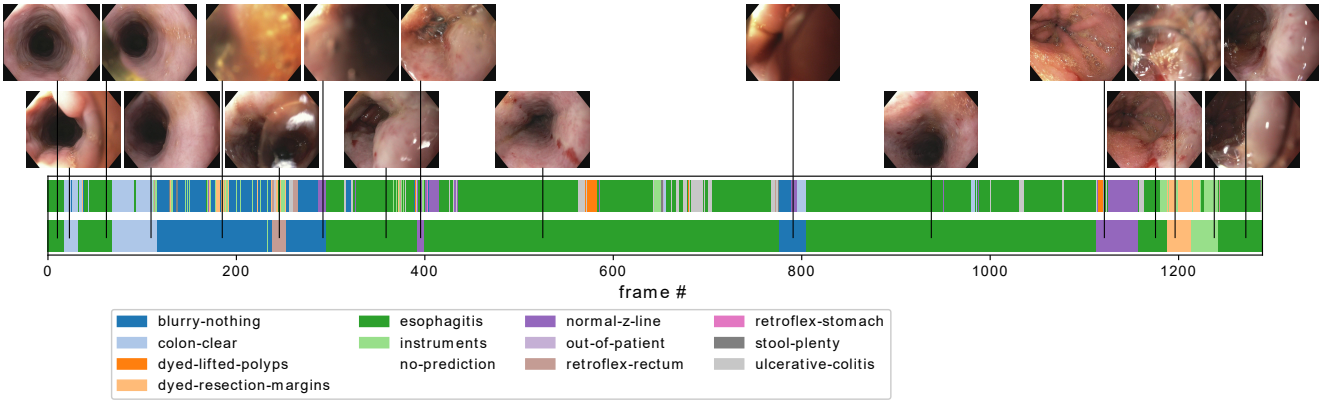

Figure 3: Resulting classifications on a per frame basis for 3e3a7ac0-4244-46cc-89a1-44ce84dd1ccf.avi. The upper bar shows the raw classifications for every frame within the video. The bottom bar shows the classification smoothed over a time period of 30 frames. We also depict one example frame for each smoothed section within the video.

that mostly contributed to the classification outcome, we generated a detailed summary of each video. We depict one such report combined with main findings and a brief summary in Figure 2.

6 DISCUSSION AND FUTURE WORK

We presented an architecture using a DCNN to predict abnormalities and diseases from GI tract images. To improve our classifications, we employed augmentation and examined different CNN feature extractors to find models that perform best given two constraints: That is inferring the best possible predictions and to return the predictions as fast as possible while not sacrificing too much detection accuracy. In addition, we expanded our architecture to automatically generate a detailed report for a given video of a gastroscopy or a colonoscopy. This report also describes in which spatial location of the video the findings were observed.

Automatic report generation could be greatly improved in the future, if ground-truth doctors' reports become available. Then techniques like hierarchical LSTM networks [3, 6, 8] could be used to automatically generate text paragraphs in natural language. We also want to focus on improving detection accuracy by using multi-class labels, e.g., there could be a polyp together with an instrument within a given image.

REFERENCES

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

[2] Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. 2015. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association* 23, 2 (2015), 304–310.

[3] Philipp Harzig, Yan-Ying Chen, Francine Chen, and Rainer Lienhart. 2019. Addressing Data Bias Problems for Chest X-ray Image Report Generation. In *Proceedings of the British Machine Vision Conference.*

[4] Steven Hicks, Michael Riegler, Pia Smedsrud, Trine B. Haugen, Kristin Ranheim Randel, Konstantin Pogorlov, Håkon Stensland Kvale, Duc-Tien Dang-Nguyen, Mathias Lux, Andreas Petlund, Thomas de Lange, Peter Thelin Schmidt, and Pål Halvorsen. 2019. ACM MM BioMedia 2019 Grand Challenge Overview. In *Proceedings of the ACM International Conference on Multimedia (ACM MM'19)*. ACM.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[6] Baoyu Jing, Pengtao Xie, and Eric Xing. 2018. On the Automatic Generation of Medical Imaging Reports. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2577–2586. http://aclweb.org/anthology/P18-1240

[7] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980

[8] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 3337–3345.

[9] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Carsten Griwodz, Dag Johansen, Concetto Spampinato, Mario Taschwer, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. Nerthus: A Bowel Preparation Quality Video Dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 170–174.

[10] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. Kvasir: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 164–169.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4510–4520.

[12] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. 2015. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging* 35, 2 (2015), 630–644.

[13] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.