Polyp Localization for Colorectal Cancer Diagnosis

Miguel Del Rio DRMIGUEL@MIT.EDU *Department of EECS*
Massachusetts Institute of Technology
Cambridge, MA, United States

Austin Wang AUSTINW@MIT.EDU *Department of EECS*
Massachusetts Institute of Technology
Cambridge, MA, United States

Elaine Xiao EYXIAO@MIT.EDU *Department of EECS*
Massachusetts Institute of Technology
Cambridge, MA, United States

Abstract

In our work, we attempt to provide tools to aid gastroenterologists in the space of colorectal cancer. We employ machine learning, specifically neural networks, which have found significant success in applications around image and video analysis, natural language processing, and other fields. Colorectal cancer is typically screened using colonoscopies, in which gastroenterologists look for polyps through a real-time video feed in the colon. As accurate detection of polyps is essential to preventing onset of colorectal cancer, we use neural networks to provide a tool to doctors to assist with polyp classification and detection. Using VGG-19 and Resnet-50, we achieve a 0.999 AUC on image-level polyp classification on the Kvasir dataset and high PPV and NPV values. We also try patch-based neural network models for classification and localization as well as an assortment of neural networks specifically for localization, though we are unable to achieve as significant results for these models.

1. Introduction

Colorectal cancer is one of the most frequently diagnosed cancers and the third leading cause of cancer-related deaths in the United States. Colorectal cancer is usually screened for through colonoscopies, in which a gastroenterologist examines the colon through a scope, looking for pre-cancerous or cancerous polyps. While skilled gastroenterologists are good at finding such polyps, certain polyps can be hard to identify due to their physiological characteristics or simply the challenges of examining all parts of the colon within the time allotted. While the adenoma prevalence rate suggests a detection rate of over 50% in a colonsocopy, reported adenoma detection rates (ADR) can range anywhere from 7% to 53%, suggesting a need for computer-aided diagnostic tools to improve ADR rates (Urban et al. (2018)). Evidence showing that increases in ADR can significantly decrease the risk of colorectal cancer gives credibility to investing in such a technical solution to the problem. In particular, recent advances in machine learning have made a way for deployment of such models to real-world applications, particularly healthcare. In this work, we explore machine learning solutions to aid in identification and localization of polyps in images from colonoscopies, toward the long-term goal of eventually arming gastroenterologists with tools to help them more effectively prevent colorectal cancer during screening.

Technical Significance Our work builds on Urban et al. (2018), which applies similar VGG and Resnet models to classification and localization. We verified the results of this work by implementing similar models on new datasets and extended the work by considering interpretability of the results, including generating heatmap visualizations of the classification results, as well as implementing other methods for comparison, including a patch-based network and YOLO.

Dataset	Task	# polyps	# non-polyp	# sequences
Kvasir Dataset V2 (Pogorelov et al. (2017))	Classification	1000	1000	—
CVC-ClinicDB (Bernal et al. (2015))	Classification (patch-based), Localization (train+val)	612	0	29
ETIS-Larib (Silva et al. (2014))	Localization (test)	196	0	≈ 29

Table 1: Datasets used in our work

Clinical Relevance We hope that our work will contribute toward the eventual development of a tool which can assist gastroenterologists to perform colonoscopies in real-time, identifying and localizing polyps during a colonoscopy so that any polyps that they miss during their own examination would be caught by the diagnosis tool instead. Research has shown that early diagnosis and detection of polyps early on is effective for preventing colorectal cancer, but due to any number of factors such as the complexity and changing-nature of the colon shape, the adenoma detection rate of many gastroenterologists, especially among doctors with less training, could be significantly improved (Urban et al. (2018)). Thus, we believe that our contributions are essential in working toward a computer-aided diagnosis tool that gastroenterologists could use in real-time to prevent onset of colorectal cancer.

2. Datasets

We drew our image data and labels from three datasets in order to train and test our classification and localization models.

The **Kvasir dataset** (Pogorelov et al. (2017)) consists of 8000 images of varying sizes, divided into 8 classes of 1000 images containing different anatomical landmarks or phatological findings in the gastrointestinal tract. The data was collected from the Vestre Viken Health Trust (VV) in Norway and was annotated and verified by medical doctors. We used this dataset for classification, employing 1000 polyp images and 1000 non-polyp images taken from the cecum. No data is given about the specific patients from whom the images were taken, nor the count of unique patients in the dataset.

The **CVC-ClinicDB dataset** (Bernal et al. (2015)), from the endoscopic vision challenge, consists of 612 polyp images with pixel-wise ground-truth segmentations of each polyp. These images come from 29 colonoscopies. We used this dataset to train the patch-based models, as we needed segmentations in order to generate patch-level labels, as well as to train and validate localization models. As with the Kvasir dataset, no demographic or clinical data was provided with the colonoscopy images. As many of these images come from the same colonoscopies, we note that there may be high correlation among consecutive images referring to the same polyp viewed from slightly different perspectives.

The **ETIS-Larib dataset** (Silva et al. (2014)), likewise from the endoscopic vision challenge, consists of 196 polyp images, also with ground-truth segmentations. We estimate that this dataset also includes 29 colonoscopies. We used this dataset to test patch-based and localization models as per the intention of the CVC-ClinicDB and ETIS-Larib datsets in the endoscopic vision challenge. In addition to the correlation we observed in the CVC-ClinicDB dataset, we also note that the brightness and color quality of these images appears in general to be different than that of the CVC-ClinicDB dataset, making for interesting results concerning the generalizability of our training to the test set.

A summary of the datasets used in this project can be found in table 1.

2.1. Data Preprocessing

All images from the datasets above were standardized with the same methods before being fed into a model. Images were first resized to 224 by 224 pixels by 3 channels, then all pixels in the image were normalized by subtracting the mean pixel value and dividing by the standard deviation measured across the pixels in the image. This pre-processing and standardization of images allowed us to use data from the different datasets regardless of slight differences in resolution, imaging equipment settings, color, etc.

Patch-based specific data pre-processing: The patch based model used the CVC-Clinical DB as a training set and required additional steps for data pre-processing. After all the images were re-sized and normalized as described above, we split them into patches of size 32x32. We determined that a patch had a polyp by looking at the ground truth segmentation given along with each image. We then arbitrarily set some threshold, τ, to determine the minimum percent of pixels in the patch that needed to be from the polyp to be considered a positive example. We tried three different thresholds $\tau = 0.25, 0.50, 0.75$ in our experiments.

To augment the data set, images in the training set were randomly rotated at angles $\theta \in [0°, 360°)$. For some positive patch found in the image centered at (x, y), we randomly selected some shift values x_{rand} and y_{rand} uniformly at random such that $x_{rand}, y_{rand} \in (0, \frac{patch\_size}{2})$ we then observe nearby patches with centers $(x \pm x_{rand}, y \pm y_{rand})$. If any of these patches had greater than $\tau\%$ of pixels from a polyp we labeled that patch as a positive example. This strategy was crucial in increasing the size of our data set because of the huge class imbalance between positive and negative patches; we found that with a threshold of $\tau = 0.5$, only about 5% of all patches in the training set were positive examples! This made the problem very difficult to learn and necessitated data augmentation.

Localization specific data pre-processing: For polyp localization, ground-truth bounding box labels were generated from the segmentation ground-truth masks in the CVC-ClinicDB and ETIS-Larib datsets, treating the image dimensions as scaled from 0 to 1 rather than 0 to 224.

3. Methods

We implemented various methods to tackle the tasks of polyp classification and localization. The two overarching approaches that we took were to use Deep Convolutional Neural Networks (CNNs) and patch-based networks. All experiments were implemented using the Keras and Tensorflow libraries.

Convolutional Neural Networks: Many of our methods were based on deep convolutional neural networks, which have become popular for many machine learning tasks, including image- and video-level classification and localization. Each model takes in an image and uses convolutional layers, fully connected layers, pooling, and nonlinear activations such as ReLU to generate a classification or bounding box dimensions. A sample network is shown in figure 2.

Training of these networks involves computing the predictions of the model on each training example and comparing them to the associated ground truth labels in a loss function. Backpropagating the error through the model allows one to tune the weights in order to improve predictions.

In our work, we primarily focused on two base models, Resnet-50 and VGG-19. VGG-19 is a standard convolutional network with 19 convolutional or fully connected layers. Resnet-50 is a similar standard convolutional network which uses identity shortcut connections to allow for the training of deep neural networks without vanishing gradients. We explain further how we use each of these in detail below.

3.1. Classification

3.1.1. Using Deep CNNs

We used the two architectures mentioned above, VGG19 and ResNet50, as our Deep CNN models. The full VGG19 and ResNet50 architectures that we used are outlined in Table 5. We preinitialized all models' weights to Imagenet trained weights, then made all layers fully trainable during training. We split the Kvasir dataset in an 80/10/10 split

(train/val/test) before any training and testing, then only trained on the train dataset, validated on the val dataset, and tested on the test dataset. We used Adam as our optimizer, sparse categorical crossentropy as our loss function, and accuracy as our metric. During training, we reduced the learning rate by 0.2 if validation loss didn't decrease for 10 epochs. We also implemented early stopping if the validation loss didn't decrease for 50 epochs. In terms of data augmentation, we experimented with rotations, horizontal and vertical flips, and shears. We also experimented with adding dropout layers before the first two fully connected layers in each model to prevent overfitting – see the results section for more.

3.1.2. Using Patch-Based Networks

The idea of patch based classification is to take a larger image, split it into smaller sections, and perform predictions on the smaller images. These results can then be combined to predict on the larger image - in our project we used threshold voting, where the image is predicted to have a polyp if greater than some percent of patches θ are predicted to have a polyp.

The models used for patch based classification ranged from a small model (8c-8c-p-16c-16c-p-1560fc-1560fc) to using VGG19 architecture with weights from a model pre-trained on imagenet with frozen initial layers but trainable final layers to predict on the patches.

We considered having two final layers: a single unit which would predict the percent of polyp pixels in each patch and two units that would have a softmax activation and predict the probability of being in the polyp or not-polyp class. While we did perform some experiments using the regressive approach of estimating the percent of polyp-pixels in a patch, we ultimately pursued the softmax final layer and thus only present the results of models using this final layer.

The model was trained using binary crossentropy with learning rate of 0.0001.

3.2. Localization

3.2.1. Using Deep CNNs

For localization, we tested three different methods: two regression models based on VGG and Resnet, in addition to a fast method for localizing objects known as YOLO (You Only Looked Once). The architectures are given in table 6.

To implement VGG and Resnet, we appended to the base networks several fully connected layers, outputting four values representing the bounding box dimensions. We used learning rate reduction when no improvement in the validation loss was achieved after 12 epochs. We tested our code with both linear and sigmoid activation functions, the former of which acts as an identity function, and the latter of which bounds the output to between 0 and 1:

$$z = \frac{1}{1 + e^{-x}} \tag{1}$$

We found that performance was fairly comparable between the two, but as some models with linear activation functions would predict very large bounding boxes outside the bounds of the images, we opted to use the sigmoid activation function. For VGG and Resnet, the output was given as (x, y, w, h), where (x, y) is the center of the box, and (w, h) is the width and height respectively. We trained both with several different loss functions, including (1) a basic L2 loss, (2) a loss function based on the dice score, and (3) a loss function penalizing large bounding boxes. The second was tested in order to train a model that was specifically optimized with the objective of maximizing the dice score, and the third was tested because the models were outputting overly large bounding boxes to try to maximize intersection. Each model was pretrained with Imagenet weights and then trained additionally on the polyp segmentation datasets, first with the base layers frozen and then across all layers.

YOLO (Redmon et al. (2015)), as a fully convolutional network, employs a similar strategy for identifying bounding boxes as with the other methods. Rather than predicting a single bounding box for the entire image, however, it

predicts several bounding boxes for each square in an $m \times m$ grid overlaid on the image. As most of the proposed bounding boxes have low confidence scores of actually corresponding to an object, the final prediction is generated by subsequently aggregating the output boxes into a single prediction of polyp locations by non-maximum suppression. YOLO is a popular method due to its inference speed, making it a good candidate for future work in the video space.

3.2.2. Using Patch-based Networks

Using the same model we used previously to classify an image with a polyp we can attempt to identify the location of a polyp in an image. We do this by taking a test image, splitting it into patches and predicting the presence of polyps in each patch. We can then aggregate these results and create a predicted segmentation! This would mean that with a perfect model we could reconstruct the ground truth segmentation!

4. Results

4.1. Evaluation Approach/Study Design

As noted earlier, our study is broken up into two tasks: classification and localization. Classification consists of identifying whether a given frame contains or does not contain a polyp. Localization identifies with a bounding box the location of a polyp in an image. We trained and evaluated our models on each of these tasks using pre-defined training, validation, and tests splits, against our ground-truth labels per dataset. We assessed classification performance using Area Under the Curve (AUC), the area under the receiver operating characteristic (ROC) curve and a measure of separability between classes, as well as positive predictive value (PPV) and negative predictive value (NPV), defined as the following in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN):

$$PPV = \frac{TP}{TP + FP} \tag{2}$$

$$NPV = \frac{TN}{TN + FN} \tag{3}$$

We use PPV and NPV as they are more useful metrics for representing the ability to predict the likelihood of an image having a polyp. That is, PPV serves as a measure of how likely a frame identified as positive has a polyp, and NPV serves as a measure of how likely a frame identified as negative does not have a polyp.

Localization methods were evaluated based on the **dice score**, a measure approximating the overlap or intersection-over-union of two segmentations. That is, given ground truth T and prediction B, we have

$$D = \frac{2|T \cap B|}{|T| + |B|} \tag{4}$$

4.2. Classification with Deep CNNs

Using CNNs, we were able to reach a very high test AUC of 0.9994 (Table 2) with just the base VGG19 model.

Table 2: Classification using various CNN architectures and data augmentation.

Model	**AUC**	**NPV**	**PPV**
VGG19 w/data augmentation	**1.000**	**1.000**	**0.9901**
VGG19 w/data augmentation and dropout	0.9997	**1.000**	0.9804
VGG19	0.9994	0.9706	0.9898
Resnet50	0.9992	0.9800	0.9890

However, we were curious to see the reasoning behind the model's classifications, and whether or not it was actually detecting presence or absence of a polyp, or instead detecting some other confounding factor unrelated to the presence of a polyp and classifying based off of that. To get a sense of where the model was looking to classify the image, we used the Gradient weighted Class Activation Map (Grad-CAM) technique (Selvaraju et al. (2016)). Grad-CAM uses the gradients of a target class (e.g. polyp or normal) before they are passed into the final convolutional layer of the model to obtain a class-discriminative localization map showing the most important regions in the image for predicting that class. For a class c, the spatial score S^c is

$$S^c = \frac{1}{Z} \sum_i \sum_j \sum_k \frac{\partial y^c}{\partial A^k_{ij}} A^k_{ij}$$

where (i, j, k) are the dimensions of the size of the image (height, width, channels), y^c is the output of class c (before softmax is applied), and A^k_{ij} is the feature map. The spatial score is calculated by global-average-pooling over dimensions i and j for the gradient of the class output y^c with respect to the feature map A^k_{ij}. The result is then multiplied with the feature map A^k_{ij} along dimension k, then averaged over k, resulting in a spatial score map of the size of the original image. A ReLU is applied to the map and the entries are normalized to be positive, because we are only interested in features with positive influence on the class c.

When we applied Grad-CAM to our images, we found that for many images, the regions that showed up most intensely in the heatmap (i.e. that had an very positive influence on the prediction of the class) were the same. For many of the images, there was confounding information, such as a turquoise box containing the polyp extraction tool, in the bottom left corner of the image, and Grad-CAM showed that this was activating the network significantly. Thus, we decided to augment the dataset with random rotations $\theta \in [0°, 360°)$ to reduce consistency in localization of confounding data.

After retraining the model with rotations as data augmentations, we obtained an even higher AUC of 1.000. Furthermore, when we applied Grad-CAM with the retrained model to the same images that we previously saw had high intensity in the bottom left corner, we saw a drastic difference in heatmap accuracy. The Grad-CAM results for the retrained model showed consistent and highly accurate regions of high intensity overlaying the location of the polyps in the images (Image 4).

4.3. Classification with Patch-based Methods

For patch based classification, it is important to understand how well the models are performing on the patch level. We first define a baseline model that will always predict a patch does not have a polyp in it; we chose this baseline simply due to the massive class imbalance - if the model predicts a negative patch every time then the model will have a high accuracy (about 0.95).

With our baseline set, we set out to improve the PPV and AUC! Between the two architectures we considered (Section 3.1.2) we found that VGG9 with frozen initial layers performed poorly when compared to the convolutional model by a significant amount - this could be due to the frozen initial layers which never learned anything from these patches. In the future, it would be interesting to try training this model from scratch using just patches or even unfreezing the layers of the pre-trained network to see if the performance improves. Regardless, we present the results of the smaller network with different labelling thresholds as described previously (Section 3.1.2) in Table 3.

We see clearly that all methods were able to improve upon the baseline model's AUC and PPV but none were able to outperform the NPV. This is a good thing, despite our models not providing a massive improvement on the baseline, because it means the patches do have some useful information for polyp classification and a further exploration could lead to large improvements overall!

We also observe that a lower τ improved the AUC and PPV while decreasing the NPV. This is likely due to an increased percentage of positive examples when compared to the larger τ values which could help the model generalize better. After seeing this, we thought that the theoretical best we could do would be to set a threshold of $\tau = 0$

Table 3: Patch Level Classification

Model	**AUC**	**NPV**	**PPV**
Baseline + $\tau = 0.50$	0.5	**0.95**	0
CNN + $\tau = 0.25$	**0.7731**	0.8731	**0.0562**
CNN + $\tau = 0.50$	0.7599	0.8942	0.0431
CNN + $\tau = 0.75$	0.7261	0.9022	0.0413

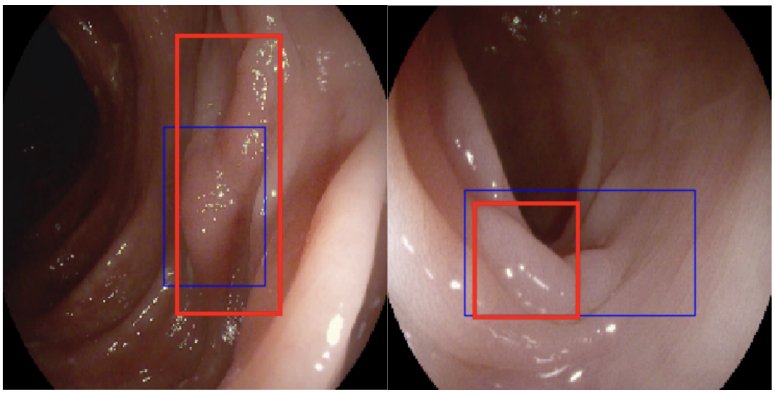

Figure 1: Sample bounding box localizations of polyps by Resnet (left) and YOLO (right).

and running a model under that labelling scheme. This can cause some problems however because this would mean patches with a single pixel would be considered positive examples but would in reality be difficult for any model to predict as having a polyp. This makes us believe that there must be some threshold value τ^* that would maximize the model's AUC/PPV beyond what we were able to achieve - in future work, we believe finding this optimal threshold would be crucial to improve model results.

4.4. Localization

Our localization results are shown in table 4. While most of our models overall performed better than baseline (which we treated as constructing a single box over the entire image), we did not get nearly as good results as the state-of-the-art in polyp detection.

Table 4: Performance on Localization Task

Model	**Dice Score**
Baseline	0.12
VGG-19	0.14
Resnet-50	**0.20**
YOLOv3	0.19
Patch-based	0.07

We generally found the best performance from the resnet-50 model. However, the best performing model optimized mainly placement of a single bounding box for all images, rather than determining the best placement of bounding box per image. We were able to achieve more variable results through custom loss functions based on square roots and penalizing on the size of the bounding box (smaller boxes have lower loss), but the dice scores were generally about the same. We hypothesize that there were either errors in our implementation or further improvements to

the models and cost functions that we could make to vastly improve performance, as results in literature were about three times higher.

The patch based method is the only one that performed below baseline—this is an indicator that the model did not learn relevant information about the polyps just using these patches. On further observations, we theorize this low score is due to the high percent of false negative patches (about 0.94) which would drag down the dice score as low as it is.

The inability of this model to locate polyps becomes clearer when we look at the predicted localization (see Appendix, fig.3). It is clearly predicting some positive patches but in reality the model is not learning much about the polyp, let alone its location.

5. Discussion and Related Work

After developing multiple models for classification, we see that the best performing models are those operating on the full image. More specifically, we found that a VGG19 trained on our data set with augmentations as described in Section 2.1. With our segmentation models, we achieved results that performed above baseline but not by a significant amount and not competitively with state of the are polyp segmentation. With further exploration using Grad-CAM, we realized that the classification model is actually looking at polyps! Using these weights, we could potentially improve our segmentation results - we leave this to future work due to a lack of time (but are excited to explore this on our own). Patch based models, we believe, have high possibility to perform better than what we present here but given our other models' performances we think further exploration in this area will be fruitless and future work should focus on full image classification/localization.

We realize that the results we have, particularly for the classification model, are extremely high - that being said it is unreasonable to expect our model to always perform so well in real world data. Despite that, given our results we truly believe that these models can be used to aide in colonoscopies and improve the probability of detecting a polyp significantly.

6. Contributions of Team Members

- **Elaine Xiao:** Implemented, trained, and evaluated the Deep CNN classification models, including heatmap visualizations.
- **Miguel Del Rio:** Implemented, trained, and evaluated the patch-based models.
- **Austin Wang:** Implemented, trained, and evaluated the Deep CNN localization models.

The source code for this project can be found at https://github.com/atwang16/sp19-6s897-colon/.

References

J. Bernal, F. J. Snchez, G. Fernndez-Esparrach, D. Gil, C. Rodrguez, and F. Vilario. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. In *Computerized Medical Imaging and Graphics*, pages 99–111, 2015.

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. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pages 164–169, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5002-0. doi: 10.1145/3083187.3083212. URL http://doi.acm.org/10.1145/3083187.3083212.

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

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

Juan S. Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. Towards embedded detection of polyps in wce images for early diagnosis of colorectal cancer. In *International Journal of Computer Assisted Radiology and Surgery*, pages 283–293. Springer Verlag, 2014.

Gregor Urban, Priyam Tripathi, Talal Alkayali, Mohit Mittal, Farid Jalali, William Karnes, and Pierre Baldi. Deep learning localizes and identifies polyps in real time with 96accuracy in screening colonoscopy. *Gastroenterology*, 155(4):1069 – 1078.e8, 2018. ISSN 0016-5085. doi: https://doi.org/10.1053/j.gastro.2018.06.037. URL http://www.sciencedirect.com/science/article/pii/S0016508518346596.

Yufeng Zheng, Clifford Yang, and Aleksey Merkulov. Breast cancer screening using convolutional neural network and follow-up digital mammography. page 4, 05 2018. doi: 10.1117/12.2304564.

Appendix A.

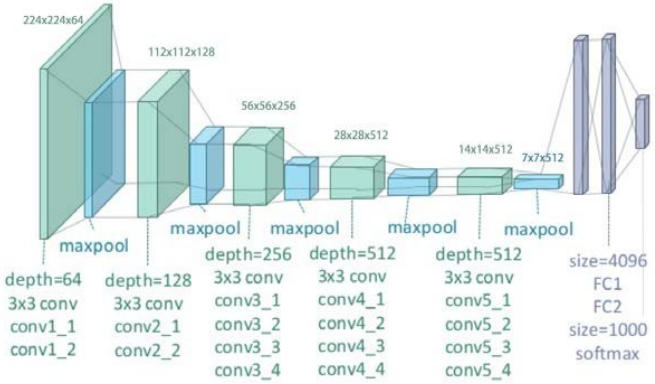

Figure 2: A sample VGG-19 network (Zheng et al. (2018)).

Table 5: Architectures of Deep CNNs used for classification.[1]

Model	**Architecture**
VGG19	64c-64c-p-128c-128c-p-$(256c)^4$-p-$(512c)^4$-p-$(512c)^4$-p-1560fc-1560fc
Resnet50	64c-bn-p-Cb-$(Ib)^2$-Cb-$(Ib)^3$-Cb-$(Ib)^5$-Cb-$(Ib)^2$-p-1024fc

Table 6: Architectures of Deep CNNs used for localization.[2]

Model	**Architecture**
VGG19	64c-64c-p-128c-128c-p-$(256c)^4$-p-$(512c)^4$-p-$(512c)^4$-gap-1024fc
Resnet50	64c-bn-p-Cb-$(Ib)^2$-Cb-$(Ib)^3$-Cb-$(Ib)^5$-Cb-$(Ib)^2$-gap-1024fc

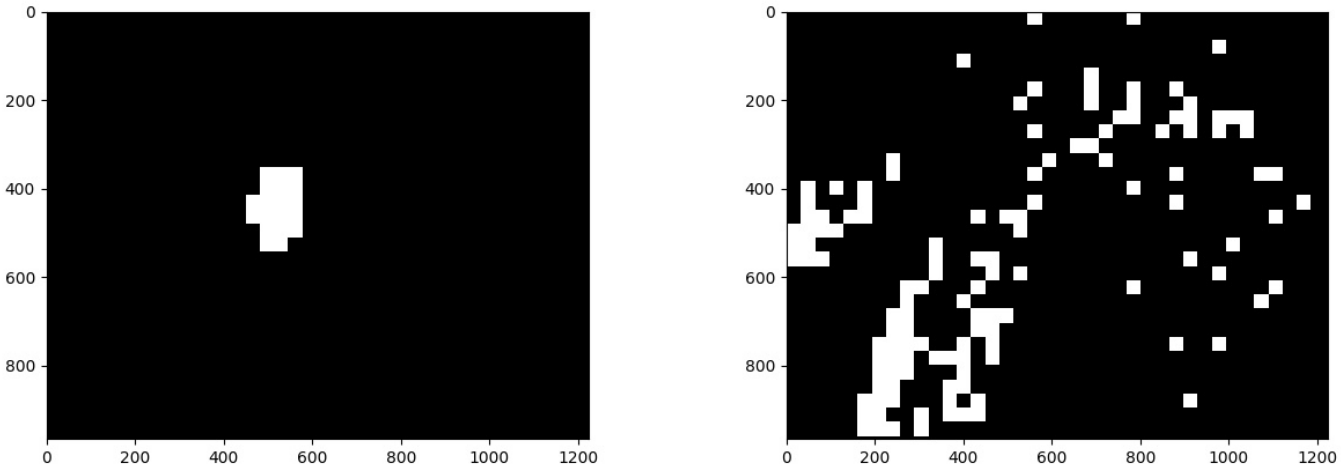

Figure 3: Left is the localization result from an idealized model that was always correct. Right is the results from our model's predictions.

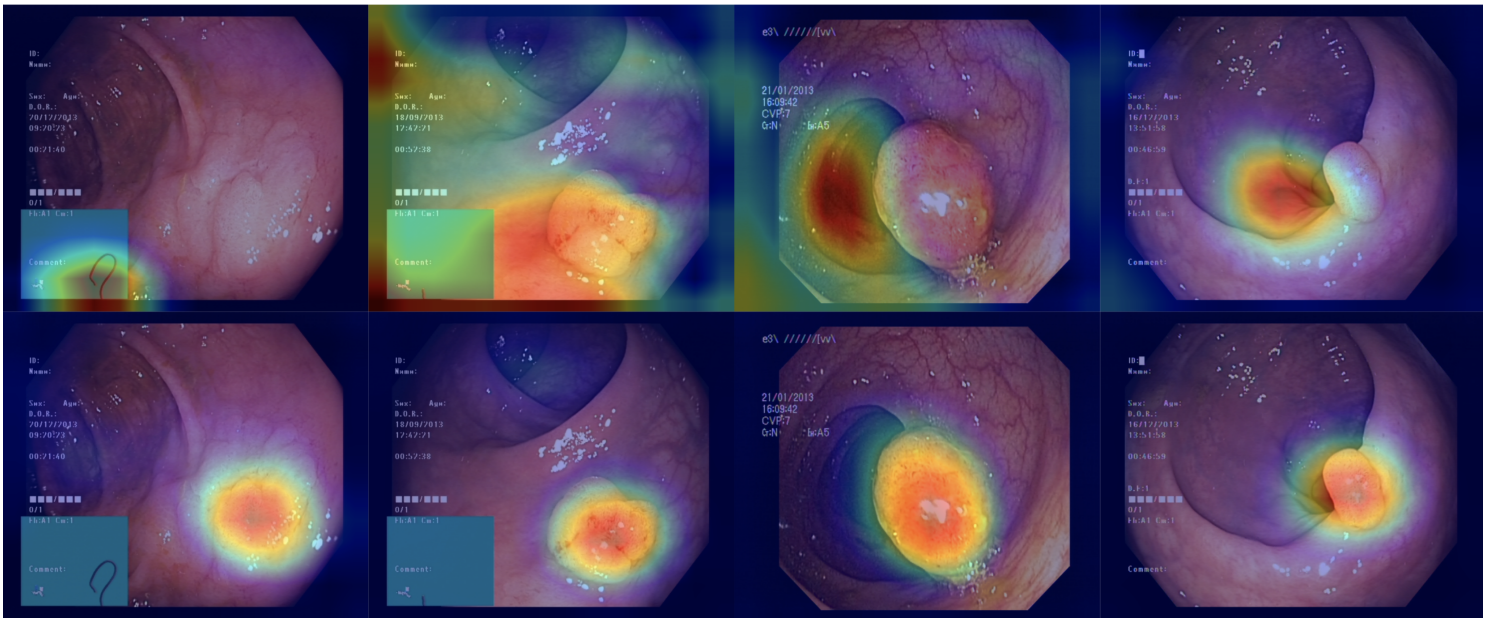

Figure 4: VGG19 Grad-CAM results (top) vs. VGG19 trained with data augmentations Grad-CAM results (bottom). From left to right: 1. Grad-CAM for the original VGG19 model focuses on the polyp extracting tool, while the retrained VGG19 model correctly focuses on the polyp. 2. The original model focuses on the bottom left corner and the text on the left, while the retrained model focuses specifically on the polyp. 3. The original model incorrectly focuses on the hole for the colon tract, while the retrained model correctly focuses on the polyp. 4. The original model focuses on a region that is slightly shifted from the actual polyp location, while the retrained model again correctly focuses on the polyp.