An end-to-end Pipeline for Automatic Polyp Detection in Colonoscopy procedures

Rajan Hari Ambrish[*, 1]

[1] *School of Computing and Augmented Intelligence, Arizona State University*

ABSTRACT

Colon Cancer initially originates as a polyp on the walls of the colon. If these are identified in the earlier stages, the risk of colon cancer can be greatly mitigated. Optical colonoscopy is the most prevalent method used by hospitals for colon cancer screening. However, These procedures are not error free as existing CAD systems have limited sensitivity and specificity. Hence, there is a lot of scope for automation in these procedures. This work aims to increase the quality of examination by alerting the colonoscopist of an improper examination and also the presence of potential polyps. We propose an end to end pipeline which assess the information content of the frames, classifies each informative frame with polyps and segments out the polyps for visualization. This can reduce the indecisiveness of the examiner during examination of certain parts of the colon with bad texture and appearance. Our experiments are designed to reveal that the proposed method achieves high sensitivity and low latency.

Keywords: Computer Vision, Machine Learning, Feature Extraction.

1. INTRODUCTION

Optical colonoscopy is a procedure in which a small camera is guided through the full length of the large intestine. It aims to scan the entire colon and detect and localize polyps, which can then be removed safely. Polyps typically take about 10 years to become cancerous, thus making colon cancer a highly preventable disease. Colonoscopy has led to a 30 % decline in colorectal cancer. However, there are many challenges associated with performing this procedure manually as humans are error prone. One particular study reports a 6 % cancer miss rate after a negative colonoscopy (False negative). This is mainly due to human factors such as lack of attention and hastiness while performing the procedure. This forms the basis of this work and provides the motivation to partially automate the colonoscopy process in such a way that it does not replace the colonoscopist, but helps perform an error free procedure. In this work we present an end to end pipeline that aims to do the following -

- Classify colonoscopy frames and being informative or non informative
- Detect the presence of polyps from the video frames, given the frame is informative, and
- Segment out the polyps from the colonoscopy frames for easy visualization purposes.

The suggested technique can be employed to both alert the colonoscopist of a improper examination and the presence of a polyp in real time during the procedure. This can significantly reduce the polyp miss rate. However, there certain challenges associated with detecting polyps:

- Polyps appear in different colors due to varying lighting conditions as shown in Fig 1
- Fig 2 shows polyps which can have large intra- and inter-morphological variations
- Visibility of texture on the polyp surface can vary based on the biological factors and camera viewing angles.
- There are lot of objects that can appear like a polyp in the colon. Few examples are shown in Fig 3.

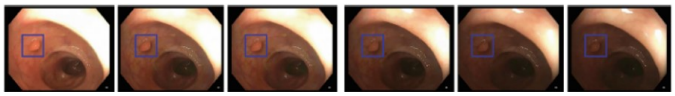

Fig. 1. Polyps under different lighting conditions

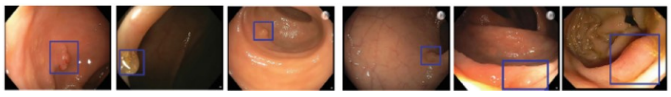

Fig. 2. Polyps with different texture and shapes

Therefore while building a classifier, it is critical to make use of all the information that can be acquired instead of

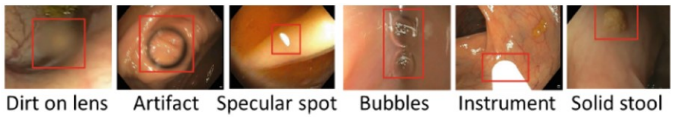

Fig. 3. Polyp mimics

extracting features that capture only a subset of the aforementioned information. We propose to use a deep learning based approach as opposed to extracting specific handcrafted features which may not adequately capture all possible information. We divide our work into three stages. In the first stage, Our method assigns each colonoscopy frame and informativeness score. A threshold can be selected as needed to classify the frames into two or more categories. In our case, we have set a threshold of 0.5. We trained and evaluated our model on 1000 colonoscopy images. The dataset had a major class imbalance, with nearly 200 frames corresponding to the blurry class and 800 frames belonging to the clear class. We provide baseline results of our initial experiments with classifying the frames. In the second part of our pipeline, we annotate each frame with the presence or the absence of a polyp and perform binary classification similar to the first stage. Finally in the third part of our pipeline, we segment out the polyps from the colonoscopy frames using architectures such as UNet. We provide good results with respect to several metrics. Our model also performs with low latency which is crucial during a real-time colonoscopy procedure. Our results for stages 1 and 2 are almost in line with human perception of the quality of the frames.

Fig 4 shows our work's end to end pipeline.

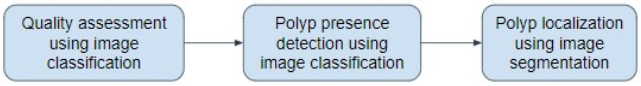

Fig. 4. Pipeline

The first stage of our pipeline is briefly described below -

- Cleaned and pre-processed the QA Polyp 2015 dataset
- Apply multiple data Augmentations to increase the size of the dataset set
- Train various Imagenet pre-trained CNNs and compared the performance
- Fine-tuned the parameters to achieve the best possible accuracy
- Calculated the confusion matrix and plotted the ROC curve
- Evaluated the area under the ROC curve and analysed the performance

The second stage of our pipeline is described below -

- Used the ASU Mayo Dataset consisting of several colonoscopy videos
- Obtained the frames (images) from the videos
- Annotated each frame based on the presence/absence of polyps
- Trained ResNet-50 model on a dataset of colonoscopy images
- Plotted ROC curve and evaluated area under ROC and analysed performance

The third stage of our pipeline, which consists of segmenting out the polyps, is briefly described as follows -

- Obtain the frames (images) from the ASU Mayo dataset
- Pre-process the frames and make it suitable for training
- Train UNet model with MobileNetv2 backbone
- Evaluate and monitor several metrics like Dice Coefficient and Recall
- Obtain the predictions of test set and analyse performance

In the remaining sections we discuss about the related work, experiments, results, conclusion, future work and contributions.

2. RELATED WORK

2.1 Automatic Assessment of Image Informativeness in Colonoscopy

There are several methods for automatic image quality assessment in colonoscopy. For the same problem, [1] proposes a system based on two key observations - blurry edges and local information content in non-informative frames. It leverages these observations to construct a difference map between a 2D DCT based reconstructed image and the original image. A histogram is then created using the difference map and a global feature vector is obtained, which is then fed into a Random Forest classifier. [2] proposed a method based on gray level co-occurrence matrix (GLCM) in the Fourier domain.

2.2 A Comprehensive Computer-Aided Polyp Detection System for Colonoscopy Videos

There are several methods for automatic polyp detection in colonoscopic videos. For the same problem, [3] proposes a tow stage method, where the first stage makes use of geometric features to generate polyp candidates and the second stages utilizes a set of deep features to effectively remove false positives. [4] proposed a method based on handcrafted texture and color descriptors like LBP and Wavelet transform. However, these techniques utilize only a subset of features such as colors and texture and subject to high variability due the colonoscopy procedure.

3. Proposed method and Experiments

3.1 Dataset

For part-1 of our work, We used the QA Polyp 2015 Dataset. It consists of over a 1000 images from colonoscopy videos. The images are annotated as being either blurry or clear.

For part-2, we used the ASU Mayo dataset consisting of several colonoscopy videos. We extracted the frames from the videos and annotated them as either containing a polyp or not containing a polyp. The annotation was done by checking the pixel values in the ground truth of the images. A frame was considered to contain a polyp (class 1) if any pixel in the ground truth had a value grater than 0.

In total, we had 600 images each for class 1 and class 2.

For part-3, we used two different datasets to include diversity in the appearance of the polyps - the ASU Mayo dataset and the CVC ClinicDB dataset. It had a total of 1230 frames and the corresponding ground truth.

3.2 Pre-processing

For part-1, The images were resized to 224×224 pixels to make it compatible with ResNet-50 architecture. The labels for each image were obtained by parsing the file containing the image Ids and the corresponding label. The dataset was split into 80-20 ratio with 20train set size of 750 images and a test set size of over 200 images.

For EfficientNetB0, the images were resized to 300×200 pixels. We did this to preserve the original aspect ratio of the images in the dataset. We further performed data augmentation to increase the size of the dataset to leverage these models learning capacity. We performed the following data augmentations -

- Horizontal and Vertical flip
- Random Rotations
- Random Brightness variations
- Random Zoom

Finally we had 1663 images with 830 clear and 833 blurry images, thus maintaining class balance.

For part-2, We calculated the frame rate as the average length of videos divided by the average number of images provided as the ground truth. This gave us a frame rate of 30 fps. We sampled the videos at 30 frames per second for optimal comparison with the ground truth images.

For part-3, the images were normalized and resized to 256×256 pixels to make it compatible for training the UNet with mobileNetv2 backbone. This size was chosen taking into account the computation complexity and loss of information when images are reduced to smaller sizes. The images were also converted to RGB format from the default BGR format for better visualization. The dataset was finally split into 80-20-20 ratio, resulting in 984 for training and 122 images each for testing and validation respectively.

3.3 Model Description and Training

We imported the ResNet-50 model pre-trained on ImageNet and froze the layers. Since we only have a medium sized dataset, we decided to fine tune the model on the dataset by adding two fully connected layers on top. Our model description can be seen in Fig. 1. It consists of two layers with 512 and 256 neurons each with the Relu activation function and a final output neuron with a sigmoid activation function. We trained our model for a total of 10 epochs. We also created a validation split of 10% from the training data for validation purposes.

We also trained EffecientNetB0 pre-trained model on our dataset in similar fashion described for ResNet-50. We trained the model for 20 epochs with a batch size of 32. All information about the Hyper-parameters used for training EfficientNetB0 can be seen in Fig 5

ID	Hyperparameter name	Value
1	Learning Rate	0.001
2	Epochs	20
3	Loss function	Binary Cross Entropy
4	Optimizer	Adam optimizer
5	Batch Size	32

Fig. 5. Hyper-Parameters (EfficientNetB0)

For part 2, we trained the ResNet-50 pre-trained model. We fine tuned the model by freezing all existing layers and

ID	Hyperparameter name	Value
1	Learning Rate	0.001
2	Epochs	20
3	Loss function	Binary Cross Entropy
4	Optimizer	Adam optimizer
5	Batch Size	32

Fig. 6. Hyper-Parameters (ResNet-50)

Hyper Parameter	Value
Learning rate	1e-4
Epochs	20
Loss function	Dice loss
Optimizer	Nadam
Batch Size	8

Fig. 7. Hyper-Parameters (UNet)

adding 3 fully connected layer on top. The model was trained for 20 epochs with a batch size of 32.

The details regarding the Hyper-parameters for ResNet-50 model for stage 2 is shown in Fig 6

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dense_1 (Dense)             (None, 256)               131328

 dense_2 (Dense)             (None, 1)                 257

=================================================================
Total params: 24,768,385
Trainable params: 1,180,673
Non-trainable params: 23,587,712
_________________________________________________________________
```

Fig. 8. Model description

For part 3, we used defined our UNet model with mobileNetv2 as the backbone. Here, mobileNetv2 acts as the encoder portion of the Unet. We designed the decoder portion similar to the encoder to obtain the final up-sampled output of size 256×256 (same as input). The bottleneck layer was of size .

We trained the Unet for 20 epochs. Several metrics were monitored for training purposes such as Dice Coefficient, Recall and Precision. Finally we evaluate our model on the test set consisting of 122 images. To further evaluate the robustness of the model, we evaluated the model on completely random images consisting of polyps to check the performance. We also calculated the time it takes for segmentation of polyps from one image or in other words the prediction latency. The various Hyper-parameters used and their values can be seen in Fig 7

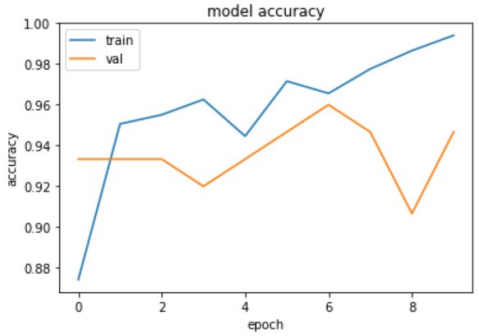

Fig. 9. Accuracy curve

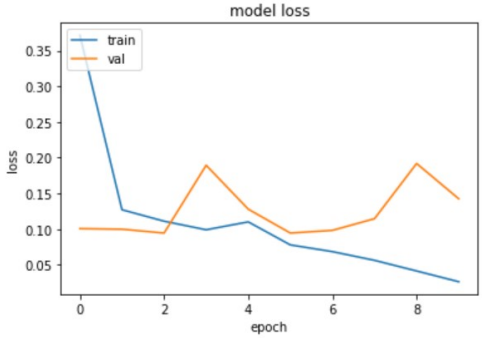

Fig. 10. Loss curve

```
[[253   6]
 [  8  52]],
```

Fig. 11. Confusion matrix

4. Results

4.1 Part-1

Experiments have been conducted as explained in the previous section, and the results for each of them have been recorded in this section. 9 shows the training and validation accuracy curve for 10 epochs. 10 shows the loss curve for the same.

The model was then evaluated on the test set consisting of 200 images. Our results are based on a threshold of 0.5 as

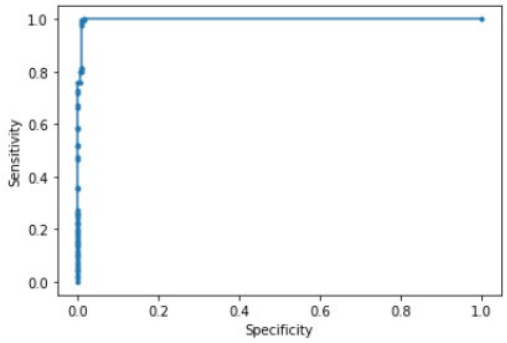

Fig. 12. ROC curve EfficientNetB0 (Informativeness classification)

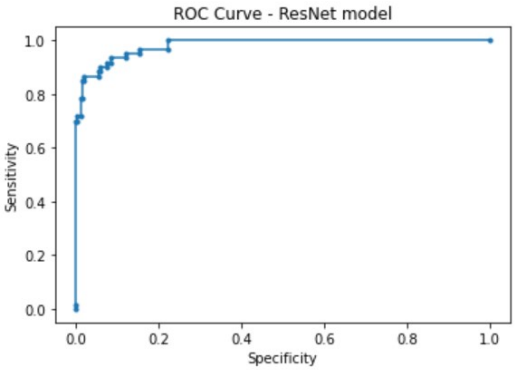

Fig. 13. ROC curve ResNet-50 (Informativeness classification)

mentioned earlier. We were able to obtain a test accuracy of 95.6%. 11 shows the resulting confusion matrix that was obtained. We got a total of 6 false positives and 8 false negatives. Our model gives a high sensitivity of 96.93% and a specificity of 89.65%. This implies that our model is very good at detecting good/informative frames and not as good while detecting bad/non-informative frames.

Fig.13 shows the ROC curve that was obtained. The resulting roc-auc-score was 0.92175.

For EfficienNetB0, we evaluated the model on a test set of 332 images. We got a test set accuracy of 97.29% which is much higher than the ResNet-50 model. Hence, we decided to incorporate EfficientNetB0 in our pipeline.

Fig 12 shows the ROc curve that was obtained for Effecient-NetB0. The resulting roc-auc-score was an almost perfect 0.9981.

4.2 Part-2

For the second stage of our pipeline, we evaluated our model on a test set of 150 images. We obtained a test set accuracy of 97.30%. 14 shows the ROC curve that was obtained. The resulting roc-auc score was an impressive 0.9977%.

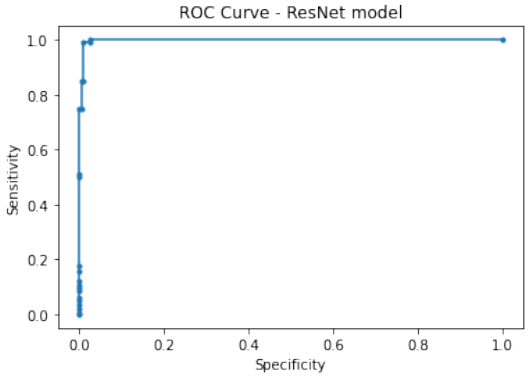

Fig. 14. ROC curve ResNet-50 (Polyp detection)

4.3 Part-3

The trained Unet model was evaluated on a test set of 122 images. We obtained a training dice coefficient of 0.8429 and a test set dice coefficient of 0.7697 as shown in Fig 15

Set	Dice Coeff	Recall	Precision
Train	0.8429	0.9151	0.9690
Validation	0.7721	0.8150	0.8380
Test	0.7697	0.7950	0.8556

Fig. 15. Results (UNet)

The frame rate in the colonoscopy videos as described before was approximately 0.035 seconds per frame. The time taken for our model to predict on a single image was 0.081 seconds. This includes the end to end procedure of obtaining a frame, normalizing, resizing, adjusting the color channels and finally calling the predict method. We have provided a few examples of our models predictions of test images for visualization purposes below.

Fig 16 shows an image without polyp. The model has predicted correctly that there is no polyp.

Fig 17 shows an image with polyp as seen in the ground truth. We can see that our model captures the polyp almost perfectly in this case.

Fig. 16. Correct prediction (without Polyp)

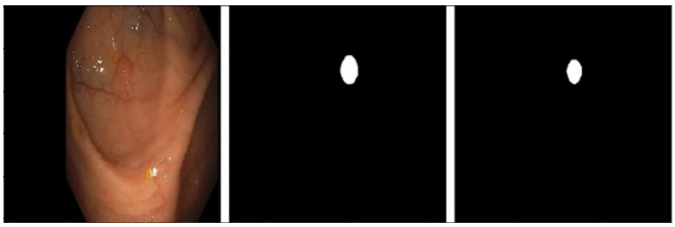

Fig. 17. Correct prediction (with Polyp)

Fig 18 shows an image with polyp, but our model has not done a good job in segmenting out the polyp. This is an example of False negative.

An example of a random image and our model's prediction can be seen in Fig 19. We can note that our model captures the polyp to an extent, but also displays a few false positive areas.

Fig 15 shows that our model has a high precision value and a slightly lower recall. This suggests that our model is almost immune to false positives and not as immune to false negatives. A thorough inspection of the predictions suggest that given the input does not contain a polyp, our model predicts no polyp with high accuracy. Whereas, given the image actually contains a polyp, our model predicts the polyp with a slightly weaker accuracy.

We also trained our UNet model for 12 epochs with the exact same experimental setup in order to visualize how the model predictions change with epochs. The results are shown in Fig **??**. It is apparent that the model trained for 12 epochs actually outperformed the model trained for 20 epochs in this particular test image. Model 20 captures the polyp just as well as the model trained for 12 epochs but it also predicts a lot of false positive regions. This may be attributed to over-fitting due to small size of the dataset.

5. Conclusion

We demonstrate the effectiveness of ImageNet pretrained models like ResNet-50 and EfficientNetB0 for classifying colonoscopy video frames. It gives performance and results

Fig. 18. Incorrect prediction (with Polyp)

Fig. 19. Prediction on random image (with Polyp)

on par with human level understanding, even with an imbalanced dataset. We also show that our model achieves a high sensitivity value.

For part 2 of our pipeline, we present nearly perfect classification AUC ROC value of 0.9977 using ResNet-50.

For part 3, we provide good segmentation results and demonstrate our model's effectiveness on random unseen images. We also provide high precision and a comparatively lower recall value which suggests that our model gives lower false positives compared to false negatives. Finally, we discuss a few areas of improvement across all 3 stages of our entire pipeline.

6. Future Work

This work provides results and accuracy in line with human perception. However, further improvements can be made. As future work, We are planning to implement different CNN architectures and compare the results to obtain the best performing model for stage-2. We propose to use CNN as a feature extractor and perform classification using random forests and support vector machine algorithms. Lastly, the threshold value for the decision can also be adjusted to improve performance. For stage -3, we plan to explore and compare the performance of various backbone models for Unet. We also plan to experiment with various segmentation models such as Attention UNet and UNet++

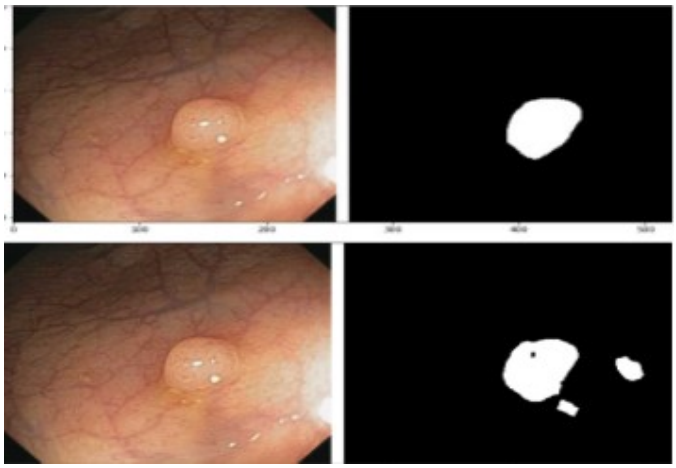

Fig. 20. Prediction comparison of 12 epochs vs 20 epochs (12 epochs at the top, 20 epochs at the bottom)

and compare their performance. We also plan to perform FROC analysis for various thresholds for the segmentation tasks. The Segmentation task can be replaced with bounding box detection tasks if it proves to be computationally more efficient and clinically useful.

7. Contributions

The following were contributed by Rajan Hari Ambrish -

- ASU Mayo dataset frame extraction and preprocessing
- ResNet-50 parameter tuning for stage-1
- Confusion matrix and explanations for stage -1
- Unet with mobileNetv2 backbone model definition and training
- Hyper-parameter tuning, latency calculations and predictions insights for Segmentation tasks.

REFERENCES

[1] *Tajbakhsh, N., Chi, C., Sharma, H., Wu, Q., Gurudu, S.R., Liang, J. (2014). Automatic Assessment of Image Informativeness in Colonoscopy. In: Yoshida, H., Näppi, J., Saini, S. (eds) Abdominal Imaging. Computational and Clinical Applications. ABD-MICCAI 2014. Lecture Notes in Computer Science(), vol 8676. Springer, Cham. https://doi.org/10.1007/978-3-319-13692-9*$_1$4.

[2] *Informative frame classification for endoscopy video," Oh, JungHwan et al. "Informative frame classification for endoscopy video." Medical image analysis vol. 11,2 (2007): 110-27. doi:10.1016/j.media.2006.10.003.*

[3] *Tajbakhsh, Nima, Suryakanth R. Gurudu, and Jianming Liang. "A comprehensive computer-aided polyp detection system for colonoscopy videos." International Conference on Information Processing in Medical Imaging. Springer, Cham, 2015.*

[4] *Alexandre, L.A., Nobre, N., Casteleiro, J.: Color and position versus texture features for endoscopic polyp detection. In: International Conference on BioMedical Engineering and Informatics, BMEI 2008, vol. 2, pp. 38–42. IEEE (2008).*