Deep Learning for Polyp Detection in Colonoscopy

[1] Chieh-Wen Chen (陳玠妏) Chiou-Shann Fuh (傅楸善)

[1] Graduate Institute of Biomedical Electronics and Bioinformatics,

National Taiwan University, Taiwan

E-mail: r08945002@ntu.edu.tw fuh@csie.ntu.edu.tw

ABSTRACT

Colonoscopy is the gold standard for polyp detection, but polyps may be missed. Deep learning technologies may assist in polyp detection.

We aim to improve the performance of a deep-learning algorithm for polyp detection in a real-world setting of routine colonoscopy with variable bowel preparation quality.

Keywords: *Colonoscopy, Deep Learning, Polyp, Object Detection.*

1. Introduction

According to statistics, the mortality rate of Colo-Rectal Cancer (CRC) ranked third in global cancer mortality in 2017. The incidence rate also placed third in the gastrointestinal tumor. Recently, the diagnosis and treatment of colorectal cancer have been highly valued.

Moreover, the major predecessor of colorectal cancer is from a colorectal polyp, and the others are from chronic colorectal inflammation. Thus, it is vital to find colorectal polyps as early as possible. Due to the different shapes of polyps, the diagnosis of polyps is full of challenges. Some small, flat polyps are easily missed and misdiagnosed.

Many researchers are dedicated to researching the automatic detection and identification of polyps by advanced methods. Currently, machine vision-based on deep learning is widely used in the field of image processing. Accurate and automatic segmentation technology for colonoscopy images can reduce the difficulty of polyp datasets acquisition. Therefore, more and more attention has been paid to the study of polyp image segmentation**[1][2]**.

The early detection and prevention of CRC is often done through regular screening. Endoscopists can easily treat small polyps that have not spread. They can even remove them before they turn into a cancerous growth.

However, there are still some limitations in the detection rate of polyps. Studies have shown that the rate of missed diagnosis of flat polyps is higher than that of the uplift type. The rate of missed polyps less than 5mm is higher than that of polyps above 5mm, and there are still many missed polyps in colonoscopy. Therefore, this study is going to develop a deep learning polyp detection system to improve the detection rate of polyps.

2. Methodology-Classification

2.1 Datasets

In this study we use 3 different polyp image datasets: CVC-ClinicDB **[3]**, Kvasir-SEG **[4]** for

training and validation; ETIS-LaribPolypDB **[5]** for testing.

CVC-ClinicDB is a database of frames extracted from colonoscopy videos. CVC-ClinicDB contains 612 standard-definition still images of 384 × 288 pixel resolution with 31 different polyps from 31 different sequences.

The Kvasir dataset comprises 1,000 images. These images were collected and verified by experienced gastroenterologist from Vestre Viken Health Trust in Norway.

In ETIS-LaribPolypDB dataset, there are 196 standard-quality polyp images from 34 different video sequences and 196 clinicians labeled ground truth images with a size of 1,225 × 966 pixels.

2.2 Data Augmentation

Deep learning models such as CNN require voluminous data to train the model without overfitting. In other words, the quality of the network training results is directly dependent on the number of datasets. This is the biggest challenge in the biomedical images. Available data are limited, and most of them are raw images without annotations.

In terms of polyp images, the acquisition of images may be hindered by a few cases, such as patient privacy limits, the need for doctors with clinical experience to label images, and the devotion of too much time to image labeling. This problem can be overcome by applying data augmentation to the input images and the corresponding ground truths.

Augmentation methods were chosen according to the appearance of the polyp images. Polyps change in shape and size, so image scaling and shear transform helped generate more data from the same image with different transformations. The images are sheared from a factor of 0.2. Polyps also appear in different locations. To encounter these locations, the images were rotated through a range between angles of (0, 90) degrees.

2.3 Data Preprocessing

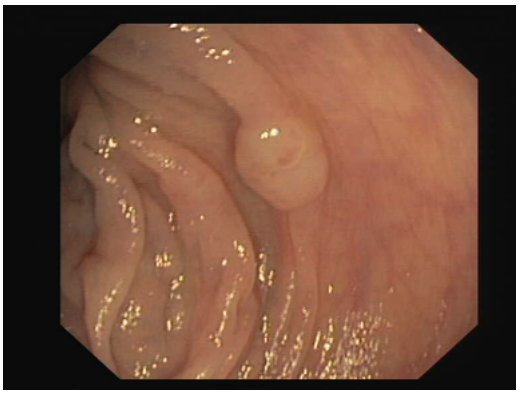

Figure 1 Original image.

Figure 2 Ground truth image.

In order to create a polyps / non-polyps images dataset, we should use the downloaded dataset and extract only the polyps and non-polyps images from all original images. This step is using original images including colorectal polyps and the ground truth images.

The new images will be saved into cropped folder. We have 2 subfolders: polyps and non_polyps.

For example, Fig. 1 is the original image, and Fig. 2 is the correspondent ground truth image, which is a localization of the polyp in the original image.

To create polyp images, we eliminate the black margin first. Second, we need to crop the polyp from original images using the white section from the correspondent ground truth images.

After extracting the polyps, we are going to generate non-polyp images. We set a minimum box 150x150 pixels to crop from images for the non-poly images. We use the same original image and we get non-polyp sections arround the polyps.

In the next step, we need to split these files into train and validation subsets for each class. We use

a dataset split percentage about 70% train and 30% validation.

Now we have a splitted dataset into train and validation subfolder with each class inside: 3,172 images in the entire dataset; 2,222 images for training: 1,111 polyps + 1,111 non-polyps; 950 images for validation: 475 polyps + 475 non-polyps.

2.4 VGG16 Transfer Learning

The most common deep learning method to process images are Convolutional Neural Networks (CNNs). By stacking multiple layers they can automatically extract important information from data and provide precise predictions regarding class affiliation or object presence in images.

Many different models have been developed but the original VGG16 was introduced by Visual Geometry Group at University of Oxford **[6]**.

Our architecture is inspired by the original VGG16 and is shown as Fig. 3.

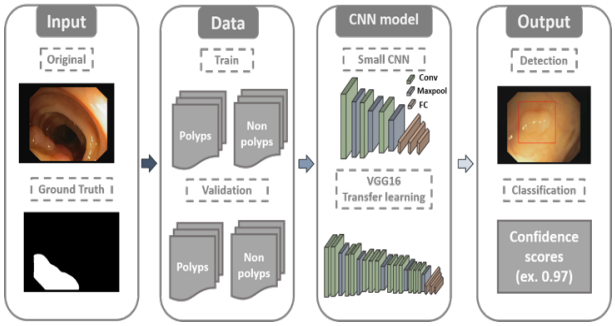

Figure 3 Framework of our research.

The VGG16 model acts as a feature extractor for the model to extract features over the entire image. It contains five CNN blocks with 3 × 3 convolutional filters and three Fully Connection (FC) layers followed after the last max-pooling layer, as shown in Fig. 4.

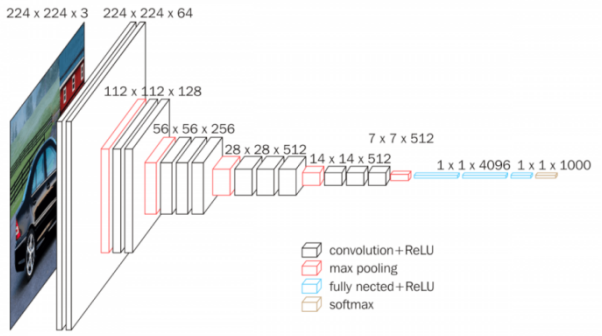

Figure 4 Architecture of VGG 16.

We apply transfer learning **[7]** using pre-trained VGG16. First, we get VGG16 pre-trained for ImageNet **[8]** without the top layer, and calculate the outputs for training and validation sets using bottom pre-trained VGG16.

Then, the saved outputs for training and validation from the bottom VGG16 will be used as inputs for the FC layer (top model) to train.

2.5 Model Fine-tuning

In the previous step we applyed the VGG16 transfer learning by training only the last FC layer while all the other convolutional blocks had the weights from the pre-trained VGG16.

Therefore, we try to implement a fine-tuning**[9]**: to train 1 or 2 convolutional blocks + FC layer.

We load the pre-trained VGG16 as the lower model, and add the top model as a FC layer.The FC layer will use initial weights from the best model obtained in the previous step.

Additionally, we freeze a number of layers (a specific number of convolutional blocks), for instance, to freeze the last convolutional block, freeze 15 layers; to freeze 2 last convolutional blocks, freeze only 11 layers.

Try to search the best model using different values for the main hyperparameters such as epochs, batch size, learning rate, momentum, and the number of layers to freeze.

Moreover, we use earlystopping if the validation accuracy is not increasing in 10 iterations.

The summary of our fine-tuning model is shown as Fig. 5.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 150, 150, 3)	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
sequential_1 (Sequential)	(None, 1)	2097665

Total params: 16,812,353
Trainable params: 15,076,865
Non-trainable params: 1,735,488

Figure 5 Summary of our fine-tuning model.

3. Methodology-Detection

3.1 Datasets

In this study we used 4 different polyp image datasets: CVC-ClinicDB **[3]**, Kvasir-SEG **[4]**, and PLOS-ONE **[10]** for training and validation; ETIS-LaribPolypDB **[5]** for testing.

PLOS-ONE consists of 758 images, which were collected from 215 patients who underwent endoscopic examinations at Sir Run Run Shaw Hospital in Zhejiang province in China from January to June 2015.

3.2 Data Preprocessing

We resize all images into 416x416 pixels, and label them with LabelImg **[11]**, which is a visual GUI-software for marking bounded boxes of objects. Those annotations are saved as .xml files.

However, YOLOv4 **[12]** training format are .txt files. Therefore, we have to create .txt file for each .xml file and the content are listed below:

<object-class> <x_center> <y_center> <width> <height>

3.3 YOLOv4 Transfer Learning

A modern detector is usually composed of two parts, a backbone which is pre-trained on ImageNet and a head which is used to predict classes and bounding boxes of objects.

In this step, we use YOLOv4 **[12]** for object dection.

YOLOv4 consisits of backbone, neck and head, which are CSPDarknet53 **[13]**, SPP **[14]** & PAN **[15]** and YOLOv3 **[16]** respectively.

The reason why we choose YOLOv4 is because it runs twice faster than EfficientDet with comparable performance. Moreover, it improves YOLOv3's AP and FPS by 10% and 12%, respectively, as shown in Fig. 6.

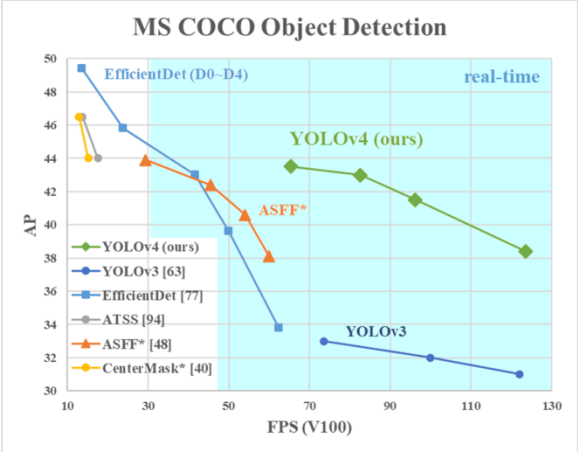

Figure 6 Comparison of the proposed YOLOv4 and other state-of-the-art object detectors.

First, we downloaded the pre-trained weights file **[17]** for the convolutional layers.

Second, we create file yolo-obj.cfg and change the number of batch size, subdivisions, classes and filters.

With all these preparation done, we started training and stopped when 6,000 iterations are finished.

4. Expected Results

4.1 Model Training Curve

For classification, after 400 epochs, our VGG16 transfer learning model achieve 99% training accuracy while validation accuracy only 91%, which is an obvious situation of overfitting. The traing curve is shown as Fig. 7.

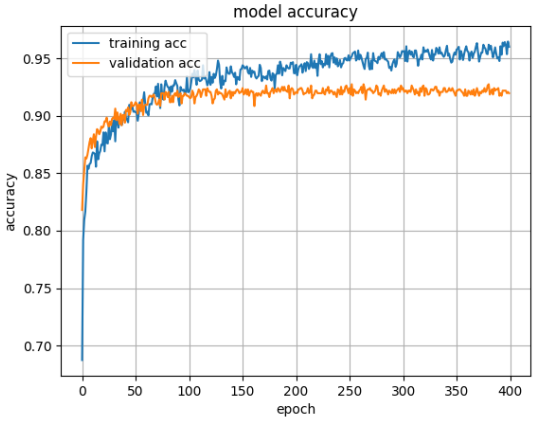

Figure 7 The training curve of our VGG16 transfer learning model.

As for the fine-tuning model, which we train the last 2 convolutional block and the fully connected layer, increases validation accuracy from 91% to 96%, as shown in Fig. 8. In other words, the overfitting is solved.

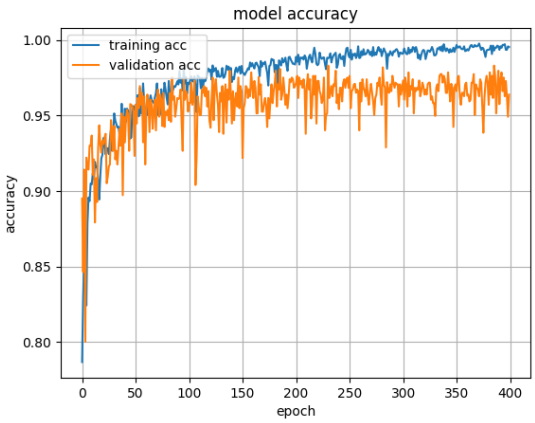

Figure 8 The training curve of our fine-tuning model.

For object detection, Fig. 9 shows the traing curve of our YOLOv4 transfer learning model . We achieve 91% accuracy after 6,000 iterations.

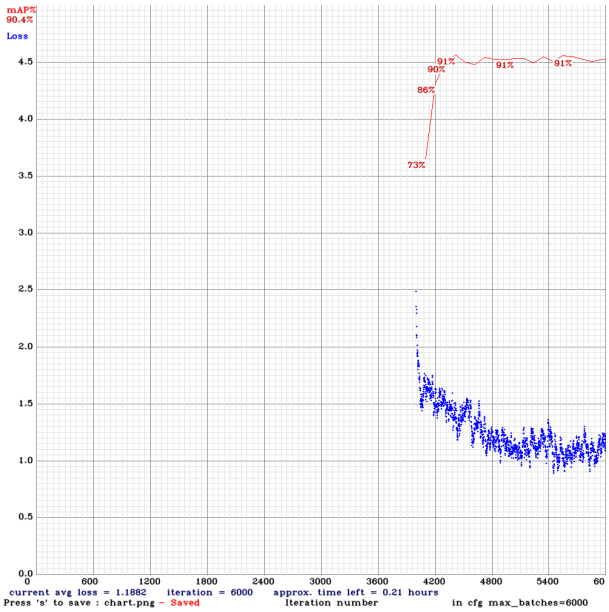

Figure 9 The training curve of our YOLOv4 transfer learning model.

4.2 Prediciton Image

The predition image will detect the location of polyp and show the probability that the polyp was found correctly, as shown in Fig. 10.

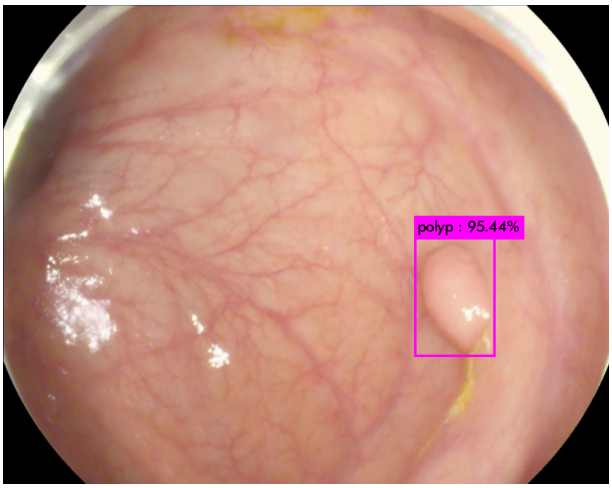

Figure 10 The prediction image that locates the polyp and shows the confidence score of this detection.

As the ground truth image shown in Fig. 11, we can find that the localization of our model is precise.

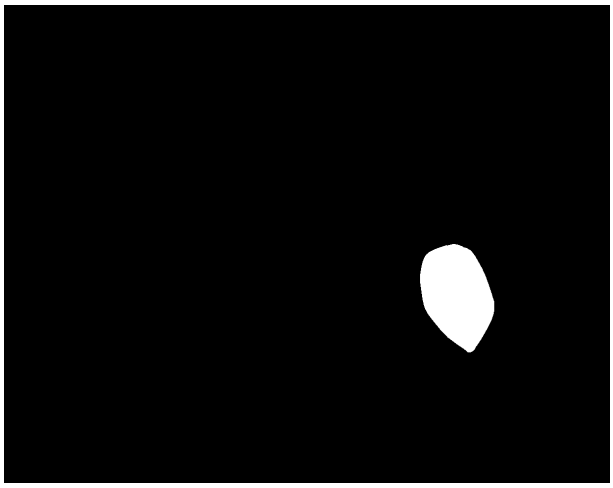

Figure 11 The ground truth image that shows the position of the polyp.

4.3 Real-time Detection Video

For real-time detection, FPS (Frames Per Second) plays an important role. Usually, the minimum permissible speed value of YOLOv4 is from 30 FPS or greater for real-time systems.

We test our YOLOv4 transfer learning model with colonoscopy video **[18]** found online, as shown in Fig. 12.

The FPS is about 45, which is optimal for real-time object detection. As we can see, the model can predict the position of polyp and show the prbability that the polyp was found correctly.

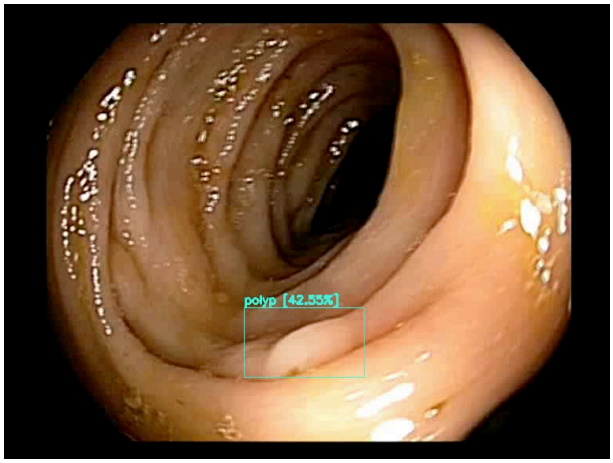

Figure 12 The colonoscopy video tested by our YOLOv4 transfer learning model.

Furthermore, our model is available to found more than one polyp in the same frame, which is shown in Fig. 13.

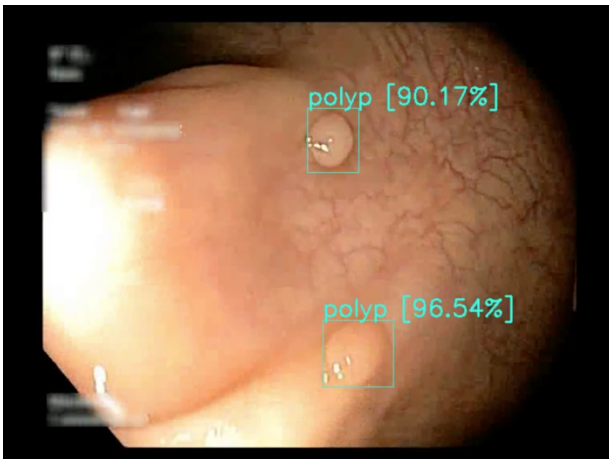

Figure 13 Our YOLOv4 transfer learning model can find more than one polyps in one frame.

Our YOLOv4 transfer learning model also perform well on Narrow Band Imaging (NBI), which is an imaging technique for endoscopic diagnostic medical tests, where light of specific blue and green wavelengths is used to enhance the detail of certain aspects of the surface of the mucosa, allowing for blood vessels improved visibility and in the improved identification of other surface structures.

Hence, no matter what kind of endoscopy endoscopists use, our YOLOv4 transfer learning model is able to find the location of polyp.

The NBI colonoscopy result is shown in Fig. 14.

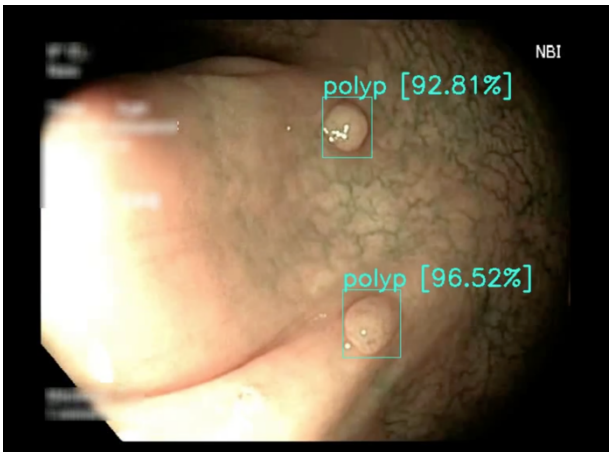

Figure 14 The NBI colonoscopy video tested by our YOLOv4 transfer learning model.

4.4 Real-time Detection Video with Sound Effect

We aim to design a real-time automatic polyp detection system providing a simultaneous visual

notice and sound effect on polyp detection, when the probability is higher than 90%.

In this way, endoscopists are allowed to focus mainly on the main monitor during the procedure and was forced to look at the system monitor by the sound effect, the output video link is shown in **[19]**.

4.5 Conclusions

Ideally, a deep learning polyp detection system, with performance close to that of expert endoscopists, could assist the endoscopist in detecting lesions that might correspond to adenomas in a more consistent and reliable way than a human assistant.

To sum up, it is available to use our VGG16 transfer learning model to classify whether this colonoscopy image contains polyp or not.

As for real-time polyp detection, our YOLOv4 transfer learning model achieves high quality, and convincing object detection results obviously.

Our deep learning polyp detection system can alert doctors to abnormalities in real time during colonoscopy, conducing to early detection of the disease.

5. References

[1] Yamada, M., Saito, Y., Imaoka, H. et al. Development of a real-time endoscopic image diagnosis support system using deep learning technology in colonoscopy. Sci Rep 9, 14465 (2019).

[2] Wang, P., Xiao, X., Glissen Brown, J.R. et al. Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. Nat Biomed Eng 2, 741–748 (2018).

[3] Bernal, J.; Sánchez, F.J.; Fernández-Esparrach, G.; Gil, D.; Rodríguez, C.; Vilariño, F. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. Comput. Med. Imaging Graph. 2015, 43, 99–111.

[4] Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Dag Johansen, Thomas de Lange, and Håvard D. Johansen, Kvasir-SEG: A Segmented Polyp Dataset, In Proceedings of the ternational conference on Multimedia Modeling, Republic of Korea, 2020.

[5] J. Silva, A. Histace, O. Romain, X. Dray and B. Granado, "Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer", International Journal of Computer Assisted Radiology and Surgery, vol. 9, no. 2, pp. 283-293, 2014.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *Int. Conf. on Learning Representations*, 2015.

[7] Bengio, Y. (2011). Deep learning of representations for unsupervised and transfer learning. In JMLR W&CP: Proc. Unsupervised and Transfer Learning.

[8] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[9] N. Tajbakhsh et al., “Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?,” IEEE Trans. Med.Imaging, vol. 35, no. 5, pp. 1299–1312, 2016.

[10] Zhang X, Chen F, Yu T, An J, Huang Z, Liu J, et al. (2019) Real-time gastric polyp detection using convolutional neural networks. PLoS ONE 14 (3): e0214133.https://doi.org/10.1371/journal.pone.0214133

[11] LabelImg.https://github.com/tzutalin/labelImg

[12] Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

[13] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. CSPNet: A new backbone that can enhance learning capability of cnn. Proceedings of the IEEE Conference on Computer Vi- sion and Pattern Recognition Workshop (CVPR Workshop), 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analy- sis and Machine Intelligence (TPAMI), 37(9):1904–1916, 2015.

[15] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768, 2018.

[16] Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement.

[17] Pre-trained YOLOv4 weights file. https://drive.google.com/file/d/1JKF-bdIklxOOVy-2Cr5qdvjgGpmGfcbp/view

[18] Colonoscopy video online dataset. http://www.depeca.uah.es/colonoscopy_dataset/

[19] The colonoscopy video tested by our YOLOv4 transfer learning model. https://drive.google.com/file/d/1nx6AwngMqL8GU7mZWVWENvl5B4hp1Bgj/view?usp=sharing