Abstract

We first compare several state-of-the-art object detection systems and decide to construct our colon polyps detection pipeline on top of YOLOv3 for its end-to-end architecture, real-time performance, less false positive errors and better generalizability. We transfer YOLOv3 model to our application by randomly initializing and retraining the weights of the three detector layers while fixing all other layers. Our system achieves real-time performance at 32.6 fps on a Titan X GPU, and attains 75.26% recall and 70.87% precision. When regarded as a classifier, our system performs well with 88.30% recall and 86.46% precision in distinguishing images that contains colon polyps from normal images. However, owing to shortage of training data, the network architecture fails to extract deep features of colon polyps. As a result, our system struggles to precisely predict the exact locations of colon polyps and sometimes misclassify other kinds of polyps as colon polyps. The codes are available on my GitHub repository `https://github.com/Bogerchen/ColonPolyps_Detection`.

Keywords: *deep learning; transfer learning; colon polyps detection; YOLOv3*

摘要

本次研究的目标是建立结肠息肉实时检测系统。本文首先比较了当前几种先进的目标识别系统。其中，YOLOv3 模型实现了端到端的设计结构、具备实时检测的能力、更少发生假阳性错误、且泛化能力更强。因此，我们认为YOLOv3 是最适合我们应用场景的模型。我们基于YOLOv3模型，运用迁移学习的知识，随机初始化并重新训练三个检测层的参数，并同时固定其他层的结构和参数。在一块Titan X 的GPU 上检测时，我们的系统达到了32.6帧每秒的检测速度，即实现了实时性。在测试集上召回率为75.26%，准确率为70.87%。我们的模型分类效果较好，用于分类正常图片和含有结肠息肉的图片时，其召回率为88.30%、准确率为86.46%。但由于训练数据缺乏，我们的模型在提取特征的能力上尚有所欠缺，使得系统有时不能精准的找到病灶的精确位置或将其他类型的息肉错判为结肠息肉。论文的代码放在我的GitHub库上`https://github.com/Bogerchen/ColonPolyps_Detection`.

关键词：深度学习；迁移学习；结肠息肉检测系统；YOLOv3

Contents

1 Introduction

1.1 Deep Learning and Transfer Learning

In the recent years, research of deep learning has made remarkable progress so that in some applications deep learning now reaches or even surpasses human performance. As a result, more and more industry domains start to take in deep learning techniques so as to assist human beings in complex tasks or even take the place.

Progress in computer vision especially object detection provides good opportunities for construction of Intelligent diagnosis and treatment system. For one thing, even an experienced doctor can only see very limited medical images for his whole career, while a neural network takes a large amount of images as input ,thus can obtain far more experience in characteristics of certain disease and unsurprisingly diagnoses more precisely than doctors. For example, Zhang Kang et al. [1] takes approximately 110,000 labeled optical coherence tomography images into their AI diagnostic system, and correctly distinguishes OCT images of diabetic macular edema, drusen and normal retina in 30 seconds. Meanwhile, the accuracy rate, specificity rate and sensitivity rate are all over 95%, which are close to the performance of human experts. For another, computers are objective and more powerful in capturing the differences among pixels and observe more details of the images. Hence, doctors can benefit a lot from the information provided by AI systems to make more accurate and stable diagnosis and decision.

Nowadays, AI systems are used to assist doctors in making diagnosis decision instead of replace them. One can imagine that the development of artificial intelligence can boost medical research. And in the near future, it’ s possible that AI systems overtake human doctors and take the place in some medical domains.

However, building a deep learning model always requires massive training data while in medical science researchers usually have very limited labeled data. As a result, it’ s unpractical to build and train a deep convolutional neural network from scratch. [2] An efficient and practical solution to the lack of data is transfer learning. PF Felzenszwalb et al. [3] used a convolutional neural network that pretrained on the ImageNet. In training, weights of the lower layers are fixed as the transferring layers and the subsequent fully connected layers are rebuilt, initialized randomly and changed during the training process. In this step, inputs are the OCT images dataset. Experiments of [3] shows that employment of transfer learning can remarkably improve the accuracy while reducing training time. There are several methods to apply transfer learning. The advantage of transfer learning is that we can make full use of the information the system learned from other large scale training data and only make little modification of the original network architecture to fit our own task. Transfer learning is so efficient that it will be more widely used in medical research that has only comparably limited data in the future.

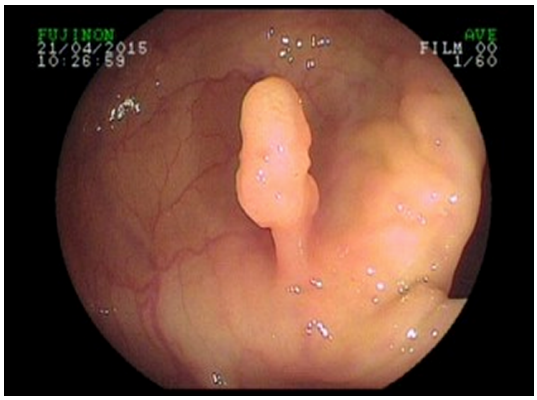


Figure 1.1: An example of colon endoscopy images.

1.2 Task Description

Our task is to build an intelligent diagnostic system that detects colon polyps in real-time. For each colon endoscopy image, our AI diagnostic system is supposed to precisely and quickly detect and localize the colon polyps. This is an object detection task for only one class.

1.2.1 Real-time Colon Polyps Detection Task

Compared to other AI diagnosis and treatment system, there are two characteristics of our task:

First, Accurate localizing. Our system is required to not only determine if there exists a colon polyp, but also localize where the colon polyps exactly are. Accurate localizing results help doctors diagnose the severity of the colonic adenocarcinoma and perform precise surgery resection.

Second, real-time performance. We work for a system that can perform real-time optical biopsy of colon polyps during colon endoscopy. Our system should be able to quickly detect abnormal objects and give an alert to assist doctors to further observe the lesion location. Therefore, our system is supposed to perform sufficiently well both in accuracy and testing speed.

1.2.2 Dataset

Our colon polyps image dataset consists of 1,164 colon endoscopy images of various size from different patients. 324 images among them are discriminated by a proficient doctor to have at least one colon polyp. And the colon polyps are localized by the doctor using LabelImg software. The results are saved as xml documents. See Figure 1.1 and Figure 1.2 for examples of the dataset.

Each xml document corresponds to an image containing colon polyp(s). The labels are:

size: size of the image, including width, height, depth.

object: detailed information of the colon polyp. The ‘name’ node means the class of the object. In this task, we have one class, that is colon polyps. The ‘bndbox’ node indicates the bounding box and (xmin, ymin), (xmax, ymax) show the top left and the bottom right

```
- <annotation>
    <folder>FFOutput</folder>
    <filename>100001~19.png</filename>
    <path>E:\FFOutput\100001~19.png</path>
  - <source>
      <database>Unknown</database>
    </source>
  - <size>
      <width>568</width>
      <height>484</height>
      <depth>3</depth>
    </size>
    <segmented>0</segmented>
  - <object>
      <name>Colon polyps</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
    - <bndbox>
        <xmin>291</xmin>
        <ymin>9</ymin>
        <xmax>432</xmax>
        <ymax>203</ymax>
      </bndbox>
    </object>
  </annotation>
```

Figure 1.2: An example of xml documents.

coordinates of the bounding box respectively.

etc.

To achieve both sufficiently high accuracy and real-time performance, we decide to construct our colon polyps detection system based on deep learning. Because of lack of labeled images, we apply transfer learning to avoid overfitting and reduce our training time.

2 Model Selection

In this section, we first display several detection systems, then conduct a deep analysis among them. At last, we will decide the most suitable system for our task.

2.1 Existing Detection Systems

Object detection is one of the core problems in computer vision. Nowadays, lots of detection models have considerable good performance, including DPM (Deformable Parts Models), R-CNN series (R-CNN, fast R-CNN, faster R-CNN), YOLO (You Only Look Once) series models (YOLOv1, YOLOv2, YOLO9000, YOLOv3), etc. Generally, detection systems repurpose classifiers to carry out detection. They accomplish the detection task through a detection pipeline which starts by extracting features from input images using convolutional networks, and then carry out classifiers and subsequently, the localizers.

Specifically, DPM [3] applies a sliding window approach and runs a classifier at evenly spaced locations over the entire image. The disjoint pipeline consists of extracting features from the current window, running classifier, predicting bounding box for high scoring regions, etc. As DPM performs detection on sliding windows separately, it fails to utilize global information of the whole image when detecting for a single sliding window.

R-CNN [4] uses region proposals instead of sliding windows to find objects in images. Selective Search generates potential bounding boxes, then a convolutional network extracts features from the bounding boxes, an SVM (support vector machine) scores the boxes, a linear model adjusts the bounding boxes, and non-max suppression eliminates duplicate detections. Each step of this disjoint pipeline should be tuned independently, resulting in a slow system which takes more than 40 seconds per image at test time. Fast R-CNN [5] and faster R-CNN [6] share computation and use neural networks to propose regions instead of Selective Search, which ends up speeding up R-CNN remarkably, but still worse than real-time.

However, unlike all those systems, YOLO [7] frames object detection as a regression problem, and uses a single neural network to predict bounding boxes and class probabilities directly from full images in one evaluation. You only look once at an image to predict whether and what objects are present and where they are. According to [7], YOLO outperforms other detection systems including R-CNN, DPM in terms of mAP (mean average precision) evaluation metric, and processes images extremely fast at test time and reaches real-time performance. Advantages of this unified model are shown in the next section.

2.2 Advantages of YOLO Architecture

- End-to-end architecture. As YOLO pipeline predicts bounding boxes and class probabilities in a single neural network, we don’t bother to tune several components of a disjoint

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 2.1: Real-time detection systems testing on PASCAL VOC 2007. Fast YOLO is the fastest detector that reaches 155 fps (frames per second). YOLO is a bit faster than R-CNN, fast R-CNN, Faster R-CNN though less accurate, and this issue is solved in the new version YOLOv3. This table is from [7].

pipeline separately. And that leads to a simple training process.

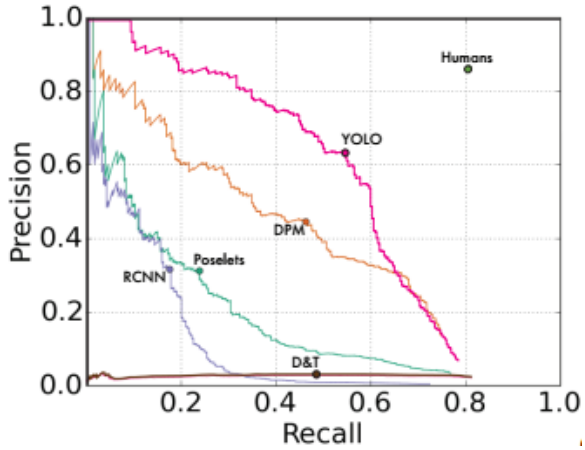
- Real-time performance at test time. YOLO model processes images in real-time at 45 fps (frames per second) on a Titan X GPU. The outstanding speed allows us to process streaming videos in real-time which in accordance with colon polyps real-time detection requirement. See Table 2.1 for comparisons of speed among YOLO and other systems. A demo of YOLO system running in real-time on webcam is present on `http://pjreddie.com/yolo/`.

- Better generalizability. In real-world applications, it's common that the test data diverges from training data. Figure 2.2 shows comparable performance among YOLO and other detection systems. On VOC 2007 detection, all models are trained on VOC 2007 data. On Picasso data, all are trained on VOC 2012 while on People-Art they are trained on VOC 2010. YOLO has good performance on VOC 2007 and degrades less than other systems when being generalized to art work datasets. In our task, We hope that our system can generalize well on the colon endoscopy images. Based on the analysis above, we think YOLO is less likely to break down when applied to our task.

- Less false positive errors. Figure 2.3 shows a detailed breakdown of results on VOC 2007. Though less accurate than fast R-CNN, YOLO is less likely to predict false positives on background. In our application, It's important that doctors make less false positive errors when applying YOLO system to detect colon polyps.

2.3 Limitations of YOLO

YOLO imposes strong spatial constraints that one grid cell can only predict one object. It doesn't matter when applied to our application as we only have one colon polyp to detect for



	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

Figure 2.2: Generalization results on Picasso and People-Art work. The left panel shows the precision-recall curves on Picasso dataset. Obviously, YOLO performs the best among all detection systems but still worse than human. The right panel displays quantitative results on VOC 2007, Picasso, and People-Art Datasets. The table indicates a better generalizability of YOLO. The figure comes from [7].

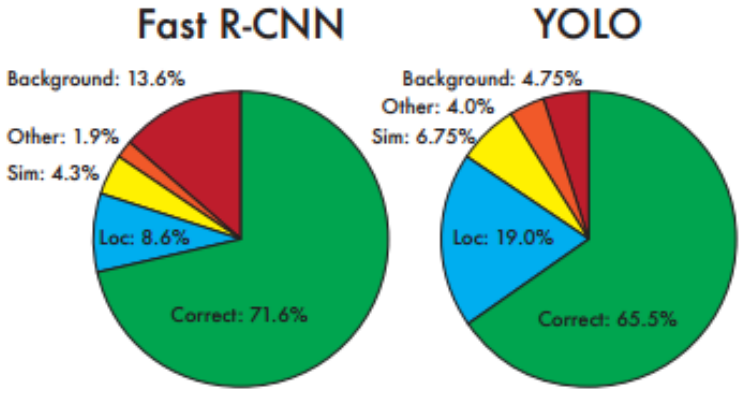


Figure 2.3: Error analysis: Fast R-CNN vs. YOLO. These charts show the percentage of localization and background errors. The figure is from [7].

about 98% of our images.

[7] points out that YOLO struggles to detect small objects such as birds in a flock. This may cause problem for colon polyps detection. Fortunately, YOLOv3 [9] carries out a wonderful solution to this problem by predicting boxes at 3 scales.

As can be seen from Figure 2.3, the main source of error of YOLO is incorrect localizations. YOLO struggles to predict precisely where the objects are present. And this problem is solved by YOLOv2 [8] by making some constraints on the bounding box center location using logistic activation.

Moreover, YOLOv2 proposes various improvements to YOLOv1 and offers a trade-off between accuracy and speed using a novel, multi-scale training method. YOLOv2 outperforms state-of-the-art methods at 40 fps like Faster R-CNN with ResNet and SSD while still running significantly faster. (See Figure 2.4)

It’ s worth mentioning that YOLOv2 applies multi-scale training method to force the network to learn to predict well across a variety of input dimensions. The input images size

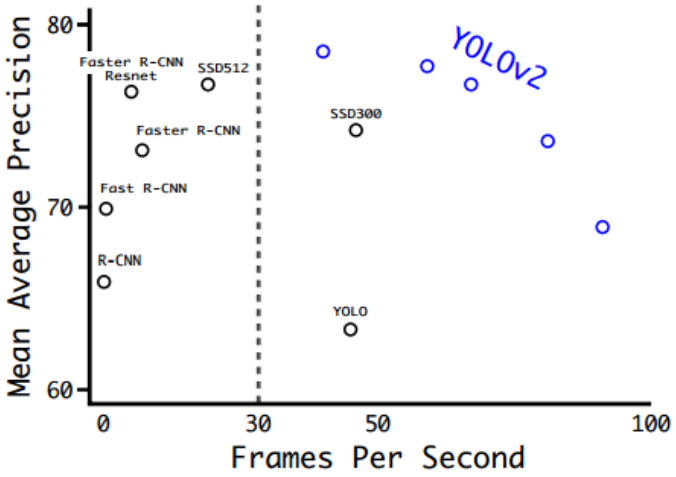


Figure 2.4: Accuracy and speed on VOC 2007 dataset. YOLOv2 performs much better than YOLO both in mAP and speed evaluation metrics. YOLOv2 runs much fast than faster R-CNN while still achieving comparable mAP. This figure comes from [8].

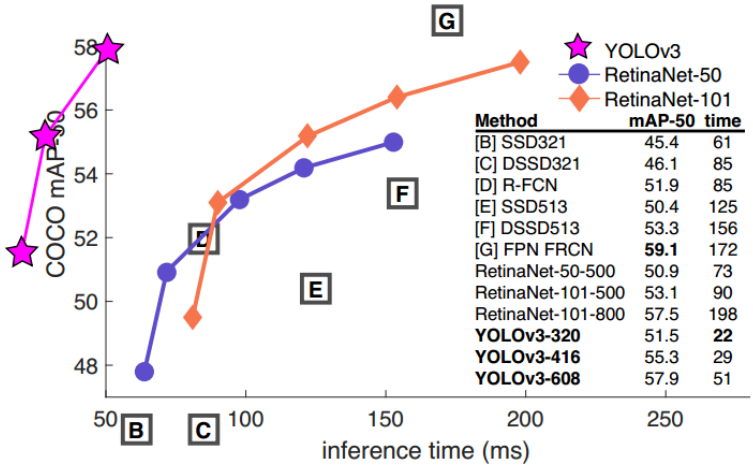


Figure 2.5: Speed and mAP-50 accuracy comparisons. YOLOv3 runs significantly than other detection methods with comparable .5AP performance. Times are from Titan X. This figure is a screenshot from [9]

changes in every few iterations rather than being fixed. This means the network is trained on images of various size and thus can perform detections at different resolutions. From this perspective, YOLOv2 fits our task so well as our colon polyps images are of various size.

YOLOv3 makes even further improvements to YOLOv2 including using a deeper, powerful network. YOLOv3 predicts 3 boxes at different scale and is now able to gain more finer-grained information of the whole image. As a result, YOLOv3 predicts small objects as well as other comparable detection systems. Figure 2.5 shows that YOLOv3 achieves comparable performance to the state-of-the-art models on COCO dataset while still fast enough.

According to the analysis above, the YOLO design enables end-to-end training and real-time speeds while maintaining high average precision. Moreover, it has several advantages over other detection systems that matches our task so well. YOLOv3 achieves state-of-the-art performance in both accuracy and speed. Hence, we decide to build our colon polyps detection system based on YOLOv3.

3 Colon Polyps Detection System

In this section, we'll introduce the architecture of the YOLOv3 and how we apply transfer learning to perform colon polyps detection. Because of our limited dataset, it is unwise to build and train our own neural network from scratch. We purpose the idea of transfer learning and build our system based on the YOLOv3 architecture. More specifically, we fix the structure and weights of the lower layers of YOLOv3, and reinitialize the weights of the last few layers including the three output layers.

3.1 YOLOv3 Architecture

YOLOv3 uses Darknet-53 framework to perform feature extraction. The network uses successive 1×1 and 3×3 convolutional layers and has some shortcut connections as well. Figure 3.1 presents the Darknet-53 framework. Figure [9] shows that Darknet-53 structure can greatly utilize the GPU, making it more efficient to evaluate and thus faster.

On top of the darknet-53 architecture, YOLOv3 adds additional convolutional layers and separately predicts 3 boxes that at each scale. (See Figure 3.1) Unlike YOLOv2, YOLOv3 excludes all pooling layers and uses convolutional layers rather than fully connected layer to be the output layers, that is, the detectors. Dimension of each detector is $3 \times (1 + 4 + C)$, where 1 stands for probability of objects, 4 denotes coordinates of the bounding box, C means the number of classes, 3 stands for three boxes. As is shown in 3.1, scale 2 detector benefits from scale 1 and scale 3 benefits from all prior computation as well as finer-grained features from earlier layers. This multi-scale design was proved to improve performance of detecting small objects.

3.2 Transfer Learning

We made a small modification of YOLOv3 architecture. That is, we changed the number of filters of the three last layers and set to 18, as we only have one single class to detect. Because of the limited data, we don't train our network from scratch and just fix the weights of all other convolutional layers except for the yolo layers, i.e. the three detectors. In training, weights of the detectors are randomly initialized and changed during training process.

3.3 Colon Polyps Detection System

We construct our end-to-end colon polyps detection pipeline on top of YOLOv3 system and detect colon polyps through a single fully convolutional network. Our system is supposed to be employed in practical application scenario, that is, assist doctors in diagnosing colon polyps in real-time during colon endoscopy. When an image sent in, our system should quickly determine if there exists colon polyps and presents the exact locations of colon polyps. And

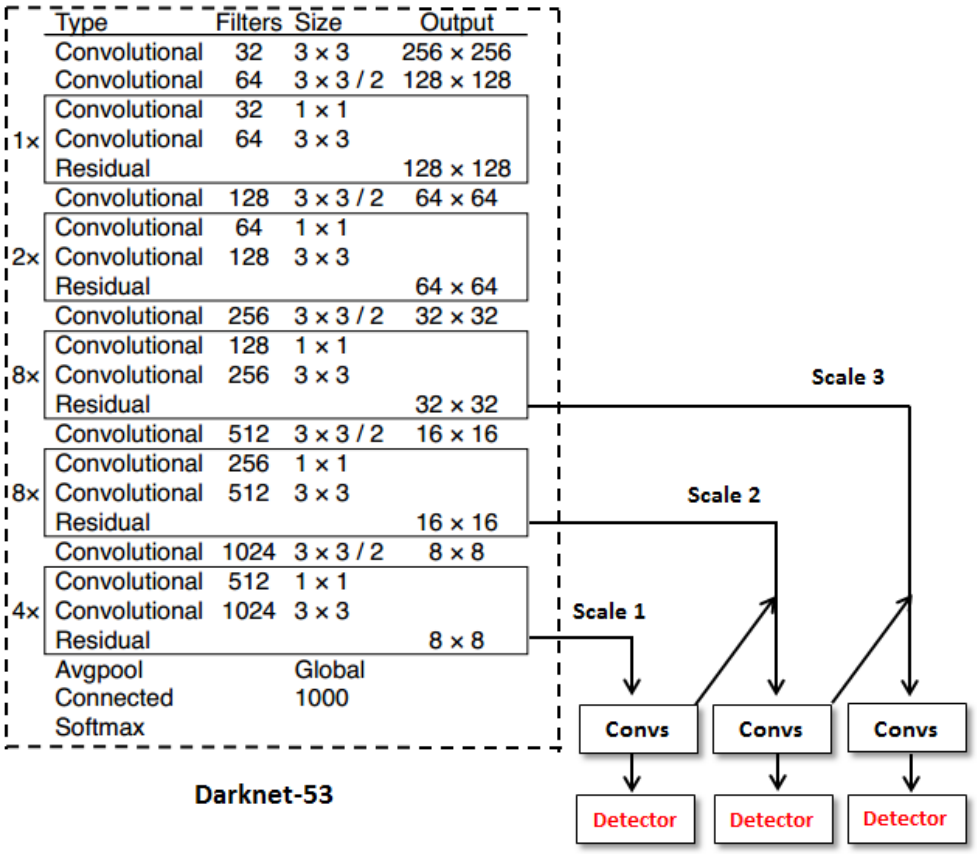


Figure 3.1: YOLOv3 architecture.

doctors can therefore decide whether to observe further into the lesion location. And that will surely improve the efficiency for both doctors and patients.

4 Experiments

We first randomly selected 224 of the 324 images that contains colon polyp(s) as the training set, and the remaining 100 images as test set. And all the 840 images that do not contain a colon polyp are used for further evaluation of our system. (See Section 4.3)

As was discussed in Section 3.2, we changed the number of filters of the three detectors, and set to 18. We just downloaded the weights of all other layers that pre-trained on ImageNet, and fixed them during training while weights of the detectors were first randomly initialized.

4.1 Training Settings

We used multi-scale training approach, batch normalization and a bunch of data augmentation including rotation, saturation, exposure, etc. Learning rate was first set to 0.001. The batch size is 64, momentum is 0.9 and 0.0005 weight decay. Our model was trained on a single Titan X for the first 2,500 batches, and later, on two Titan X GPUs. It took 15.5 hours in total to train the model.

The average loss decreases rapidly as training begins, and changes too little for about 100 batches. (See Figure 4.1) However, when we use the model that trained for only 100 batches, we get nearly zero recall meaning the model is seriously underfitting. Therefore, we decide to choose the best iterations number according to the curve of the logarithm of the average loss. According to Figure 4.1, we consider 8,500 to be the best iterations number, and evaluate our system using the weights that trained for 8,500 batches.

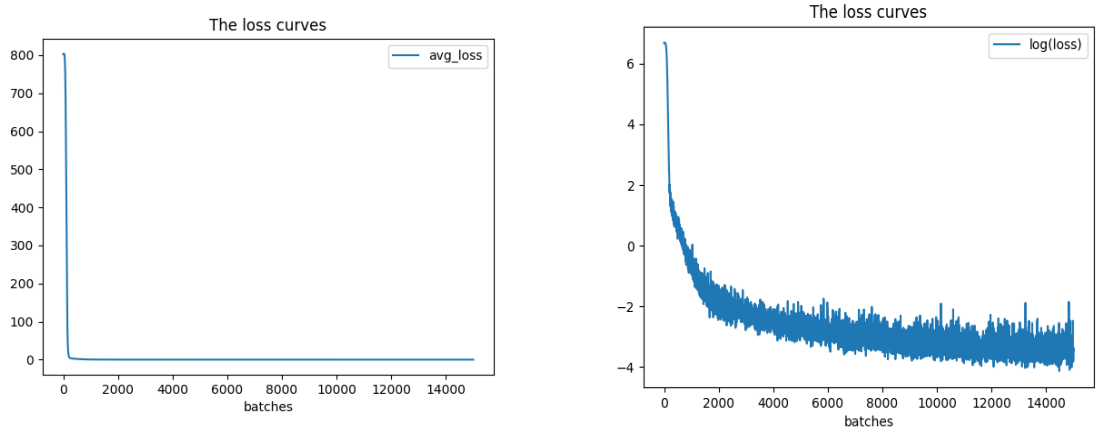


Figure 4.1: Loss curves. The left panel shows the exponentially weighted averages of loss with coefficient being 0.9. The right panel is the logarithm of the average loss.

4.2 Testing

We set the IOU threshold to be 0.3 and test our system on the test set. We get 75.26% recall and 70.87% precision on the 100 images. The scores are not that high meaning that our system needs lots of further improvements. We will discuss this problem in Section 4.4. Examples of correctly detected images are displayed in Figure 4.2.

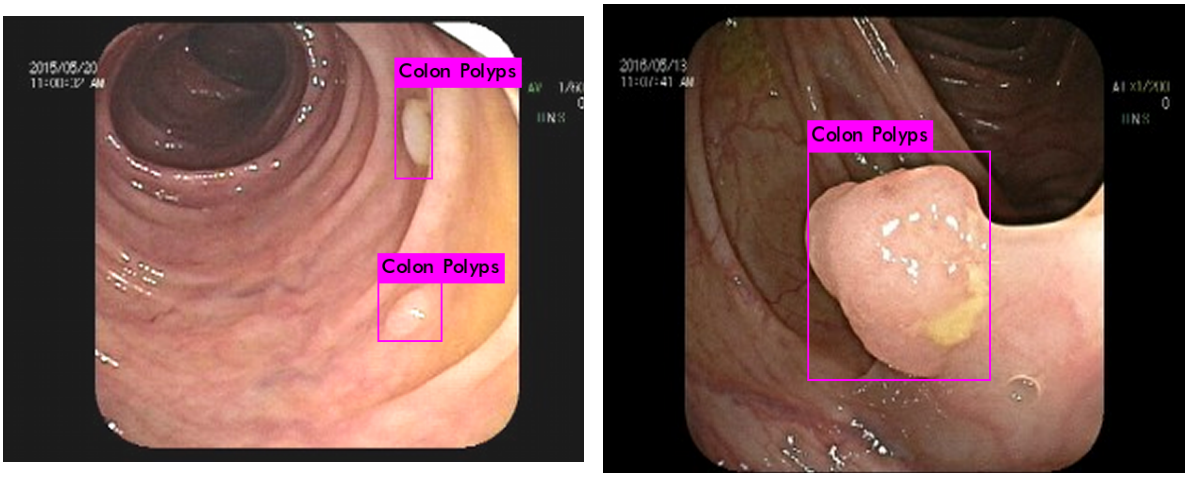


Figure 4.2: Examples of correctly detected colon polyps.

When testing on a Titan X GPU, the whole testing process costs 3.064714 seconds, i.e., 0.0306seconds per image. Therefore, our system achieves real-time performance, say 32.6 fps.

4.3 Evaluation of Classification

To further evaluate the performance of our system, we regard our system as a classifier. We wonder if our system can correctly distinguish normal images from images that contains colon polyps. In order to avoid unbalanced classification problem, we first randomly selected 100 of the 840 normal images, and combined them with the 100 images of the test set. We ran our system on each of the 200 images. Results are displayed on Table 4.1.

	Ground Truth		Total
Prediction	Normal	Colon Polyp(s)	
Normal	93	11	104
Colon Polyp(s)	13	83	96
Total	106	94	200

Table 4.1: Results of classification evaluation. recall=83/94=88.30%, precision=83/96=86.46%

At this time, the recall and precision scores are evidently higher than that of the detection procedure. We conclude that our system is capable of telling normal images from images that contains colon polyps, though struggles to figure out the exact locations of the colon polyps.

Looking deeper into the 13 false positives predictions, we find out that our system does detect some polyps though they are other kinds of polyps rather than colon polyps. All that

means is that our system needs further improvement to learn more detailed features of colon polyps. For examples see Figure 4.3.

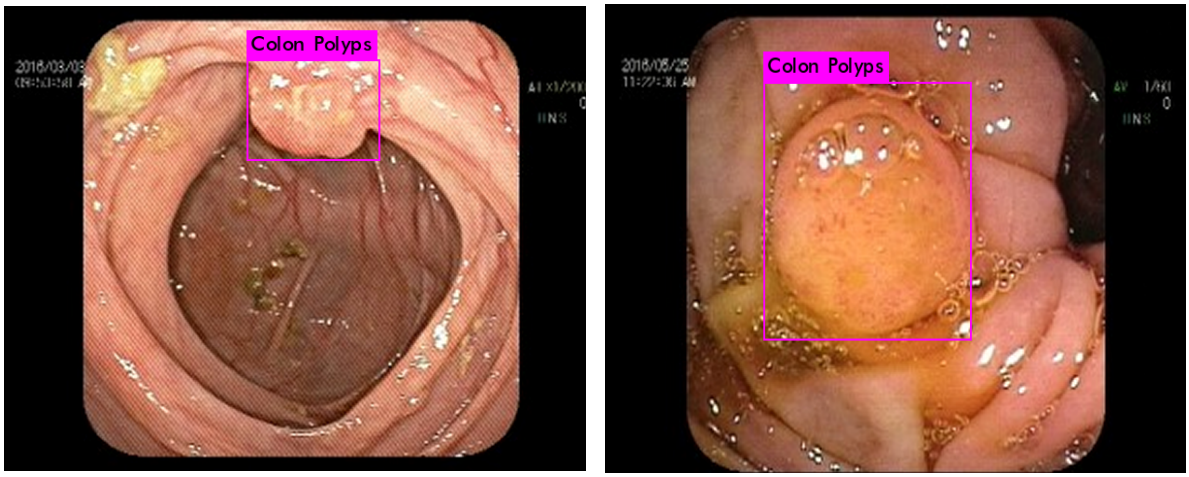


Figure 4.3: An example of false positive errors. The image in the left contains a colon polyp, while the right displays an polyp that of different category. We see that our system mistakes the polyp in the right image as a colon polyp.

4.4 Comments on Our System

According to the analysis in Section 4.2 and Section 4.3, our system struggles to discriminate colon polyps from normal polyps and determine the precise locations of colon polyps, which informs us that our system is not so good at extracting features of colon polyps. The main solution may be larger dataset considering that we now have only 224 images for training. Moreover, we can also initialize and train the weights of the three layers following which are the detectors.

5 Conclusion and Future Work

YOLOv3 is an end-to-end real-time detection pipeline using a single fully convolutional network. Advantages of YOLOv3 compared with other detection systems includes end-to-end architecture, real-time performance, better generalizability and less false positive rate. Hence, we propose YOLOv3 to carry out colon polyps real-time detection task.

We randomly initialize and train the weights of the three detection layers basing on the idea of transfer learning. Experiments show that our system performs quite well in distinguishing images of colon polyps from normal images, though struggles to figure out the exact lesion locations. Moreover, our system sometimes mistakes other polyps as colon polyps. The main reasons may be limited training set and the inappropriate transfer learning architecture.

For future work, we will collect more training data and try different network architecture to help improve the capacity of extracting deeper and more detailed features of colon polyps.

Acknowledgements

I wish to express my great gratitude to my supervisor, Teacher Huang Zhihong for his kindly guidance during my writing of the thesis, together with alumna He Cuiyi who offered several insightful suggestions on my experiments. I have also benefited from Doctor Zhong Dong for his precious colon endoscopy images dataset. Schoolmate Zhang Yikun is highly appreciated for his technical support in LaTeX.

I will always feel grateful to Teacher Ren Chuanxian. I learnt a lot in his class and restored self-confidence academically owing to his trust and encouragement. I have further been fortunate to join Yat-sen Siyuan Programme and made friends with many outstanding fellows who inspire me constantly. I owe sincere thanks to my roommate Chen Fengjie for his assistance in all his forms in the past four years. Thanks also go to my best friends Liang Peichen, Chen Junxuan, Gao Yi, Zhou Siwen who always encourage me whenever I'm depressed. Moreover, I am obliged to uncle Yang as well as teachers in Guohua for their generous help.

Finally, I want to express my greatest thanks to my mother and my family, who are my source of happiness and motivation. I am really indebted to their love, encouragement and understanding.

My best thanks to them all.

Bibliography

[1] Kermany et al., 2018, Cell 172, 1122 - 1131 February 22, 2018 2018 Elsevier Inc. https://doi.org/10.1016/j.cell.2018.02.010

[2] Shie C K, Chuang C H, Chou C N, et al. Transfer representation learning for medical image analysis[C]// International Conference of the IEEE Engineering in Medicine & Biology Society. Conf Proc IEEE Eng Med Biol Soc, 2015:711.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627 - 1645, 2010. 1, 4

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580 - 587. IEEE, 2014. 1, 4, 7

[5] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[6] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]. NIPS,2015.

[7] Redmon, Joseph; Divvala, Santosh; Girshick, Ross; Farhadi, Ali. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640. 06/2015.

[8] Redmon, Joseph and Farhadi, Ali. arXiv:1612.08242. 2016.

[9] Redmon, Joseph and Farhadi, Ali. YOLOv3: An Incremental Improvement. arXiv, 2018.