Patch-based deep learning approaches for artefact detection of endoscopic images

Xiaohong W. Gao, Yu Qian *

Department of Computer Science, Middlesex University, London, NW4 4BT, UK
x.gao@mdx.ac.uk

*Cortexcia Vision System Limited, London SE1 9LQ, UK
Yuqian_hong@yahoo.com

Abstract

This paper constitutes the work in EAD2019 competition. In this competition, for segmentation (task 2) of five types of artefact, patch-based fully convolutional neural network (FCN) allied to support vector machine (SVM) classifier is implemented, aiming to contend with smaller data sets (i.e., hundreds) and the characteristics of endoscopic images with limited regions capturing artefact (e.g. bubbles, specularity). In comparison with conventional CNN and other state of the art approaches (e.g. DeepLab) while processed on whole images, this patch-based FCN appears to achieve the best.

1. Introduction

This paper details the work by taking part of the Endoscopic artefact detection challenge (EAD2019) [1] with three tasks, which are detection (task #1), segmentation (task #2) and generalization (task #3). All three tasks are performed using the current state of the art deep learning techniques with a number of enhancement. For example, for segmentation (task #2), patch-based approached are applied. In doing so, each image is divided into 5×5 non-overlapping patches of equal sizes. Then based on the contents of their counterparts of masks, only patches with non-zero masks are selected for training to limit the inclusion of background information. Each class is trained individually firstly. Then upon the last layer of receptive fields, the features from five classes are trained together using SVM to further differentiate subtle changes between five classes.

For detection of bounding boxes (Tasks #1 & #3), while the above patch-based approach delivers good segmentations, the bounding boxes of those segments do not seem to agree well with the ground truth with Null values of IoU. Hence the state of the art models of fast-rcnn-resnet101 has been applied that gives the ranking position of 12th on the leaderboard, which is build upon tensorflow model. In addition, the models of Yolov3 by using darknet is also evaluated, which delivers detection ranks between 17 to 21 based the selection of thresholds (0.5 or 0.1).

2. Segmentation

Before training, each image undergoes pre-processing stage to be divided into 25 (5×5) small patches in equal size. As a result, the training samples have width and height sizes varying from 60 to 300 pixels. Those patches with their corresponding masks with zero content are removed from the training to level the influence of background.

For segmentation, the training applies the conventional fully connected neural network [2] built upon Matconvnet that begun with imageNet-vgg-verydeep-16 model. To minimise the influence of overlapping segments, instead of training all the classes collectively, this study trains each segmentation task individually. The final mask for each image is then the integration of five individual segmentation masks after fine tuning using SVM. In other words, the last layer of features from each model are collected first. Then SVM classifier is applied to fine tune each segmentation class to further differentiate each class.

Figure 1 illustrates the proposed approach. Firstly, each of five classes are trained on patches independently to take into account of overlapping classes. Then upon connection layer of all five classes, SVM classifier is trained to high light the distinctions between each class. This classifier will perform the final segmentation for each of five categories, i.e. instrument, specularity, artefact, bubbles, and saturation.

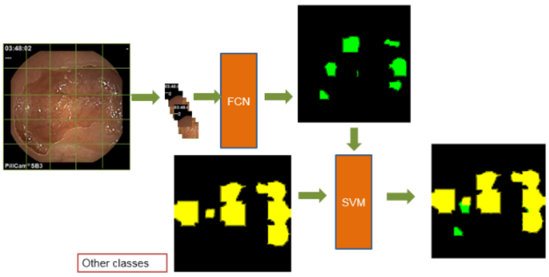


Figure 1. The steps applied in the proposed patch-based segmentation.

In addition, two other popular models are evaluated, which are fast region-based CNN [3] with resNet [4] and deepLab [5]. Table 1 presents the outcome from EAD2019 [6] leaderboard after uploading each result obtained from different deep learning models where our patch-based FCN delivers the best F2 and semantic scores.

Figure 2 demonstrates the steps taken while applying deepLab version 3 using tensorflow model [7].

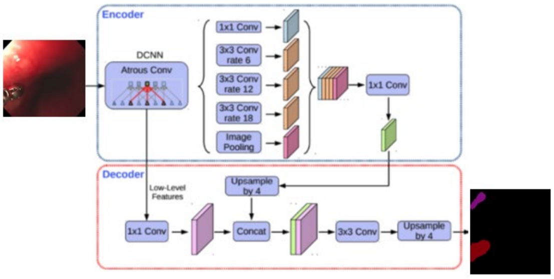


Figure 2. Segmentation applying deepLabV3 model.

Similarly, Figure 3 represents the procedures while utilizing the patch-based classification model of Caffe. The patch size is selected to be 32×32.

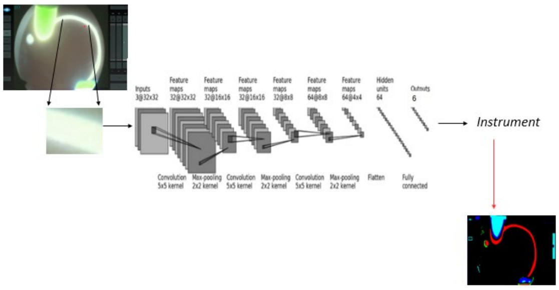


Figure 3. Caffe classification model while applying 32×32 patches.

Table 1. Competition results obtained

from EAD2019 after uploading the results.

Model	F2-score	Semantic-Score
Frcnn-ResNet101 (1000x1000 crop size)	0.2300	0.2155
deepLab (32x32 patch)	0.1638	0.1872
Patch-based FCN	**0.2354**	**0.2434**

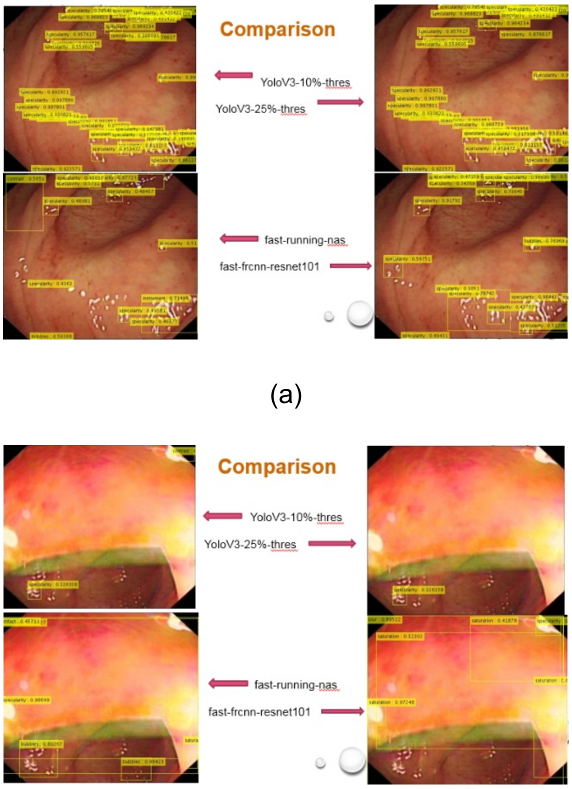
3. Detection of artefact

While the above patch-based segmentation model appears to perform well for segmentation, when it comes to detection of bounding boxes of intended segments, for some unknown reasons, the detected value of IoU_d appears to be NULL. Hence a number of existing state of the art models are evaluated due to time constraint, comprising fast running nas [8], YoloV3 [9], and fast-rcnn-resnet101 [3] and Yolo using darknet [10]. Table 2 presents the evaluation results of the above models. The Fast-cnn model with the threshold of 0.3 appear to perform the best, which is the one given on the leaderboard of EAD2019 with a rank of 12.

Table 2. The results of detection results obtained from the leaderboard of EAD2019 for each tested model (trs=threshold).

Model	IoU_d	mAP_d	Overlap	Score_d
Fast-Running-nas	0.3164	0.2425	0.2107	0.2720
Yolov3 (trs=0.1)	0.2273	0.1750	0.2331	0.1959
Yolov3 (trs=0.25)	0.2687	0.1668	0.2331	0.2075
Frcnn-resnet101 (ths=0.3)	**0.3482**	**0.2416**	**0.1638**	**0.2842**

Figuratively, Figure 4 demonstrates the comparison results between the above four models for 2 images.



(a)

(b)

Figure 4. The comparison results for the four models of fast-running-nas, YoloV3 (threshold=0.1), YoloV3(threshold=0.25), and Frcnn-resnet-101 (threshold=0.3).

Figure 5 compares the generation results (Task #3) between models Fast-running-nas (top) and fast-rcnn-resnet101 (bottom).

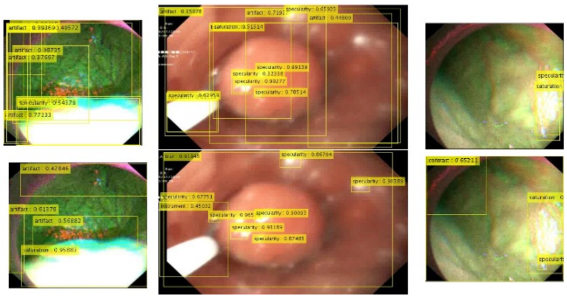


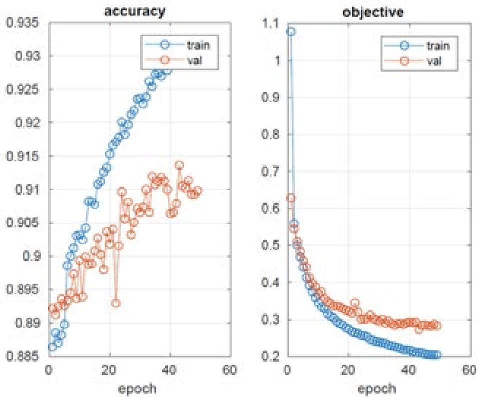
Figure 5. The comparison results of generalization task (task 3) using two models: fast-running-nas (top) and fast-rcnn-resnet101 (bottom).

4. Conclusion and discussion

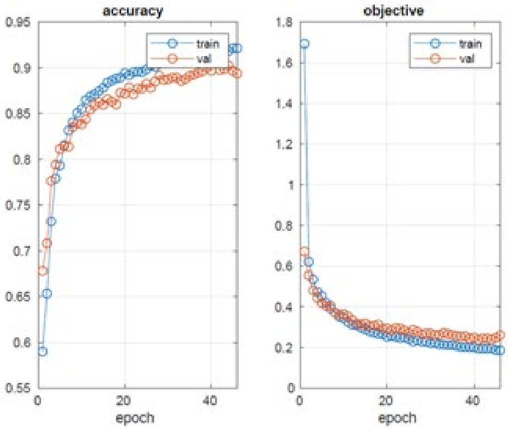
It has been a very enjoyable experience while taking part in this EAD2019 competition. Due to the late participation (two weeks before the initial deadline), implementation of several ideas could not be fully completed. However, the final position of 12 is better than expected, which is quite uplifting.

After initial evaluation of existing models (both in-house and in the public domains), it is found that, no model performs significantly better than the other. Semi-supervised approach will be recommended coupled with clinical knowledge.

Contribution includes patch-based training. While several existing models incorporate regions of interest for training, some regions appear to be overwhelmingly larger than the intended targets (>95%), hence introducing too much background information, leading to the sampling distribution substantially unbalanced. Because of the varying size of training datasets, from 300 to 1400 pixels along both width and height directions, fixed patch size may instigate under or over sampling. Hence in this study for segmentation (task #2), each image is divided into 25 equal sized patches non-overlapping, which appears to give good segmentation results. However, it is foreseen that sampling with overlapping regions collectively might deliver even better results, which will be investigated in the future. Figure 6 depicts the learning information of whole-image-based (top) and patch-based segmentation as well as whole-image-based detection (bottom).



Learning information training based on whole images



Learning information training based on **patches** – class 1

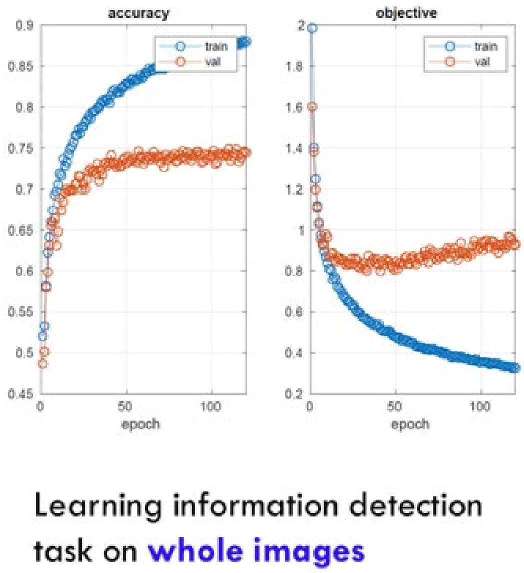


Figure 6. Learning information for segmentation based on whole image (top), patch (middle) and detection based on whole image.

Regarding to the detection tasks utilising existing models, the challenge here is to find the right threshold for the last fully connected layer of probability. Higher thresholds might miss some intended regions. However, lower thresholds tend to not only over segment but also repeat some regions a number of times. For example, to delineate one single contrast region using YoloV3 model from one test image, lower threshold (0.4) delivers to three bounding boxes, with each bigger one surrounding smaller one as illustrated in Figure 7.

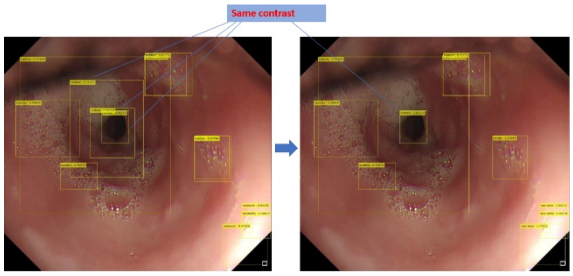


Figure 7. The impact of thresholding of model of fast-running-nas. Left: threshold=0.1; right threshold=0.3.

In summary, for medical images, medical knowledge needs to be incorporated in order to generate more accurate results.

Reference:

[1] S. Ali, F. Zhou, C. Daul, B. Braden, A. Bailey, J. East, S. Realdon, W. Georges, M. Loshchenov, W. Blondel, E. Grisan, J. Rittscher, Endoscopy Artefact Detection (EAD) Dataset, in Processdings of EAD 2019 challenge, IEEE ISBI 2019.

[2] MatCovNet FCN: https://github.com/vlfeat/matconvnet-fcn.

[3] R. Girshick, Fast R-CNN, arXriv: 1504.08083v2, 2015. https://arxiv.org/pdf/1504.08083.pdf.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv 1512.033385v1, 2015. https://arxiv.org/pdf/1512.03385.pdf.

[5] L. Chen, G. Papandreou, F. Shroff, H. Adam, Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv:1706.05587v3, 2017. https://arxiv.org/abs/1706.05587
.

[6] EAD2019, https://ead2019.grand-challenge.org/.

[7] L. Chen, Y. Zhu, G.

Papandreou, F. Schroff, H. Adam, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV2018.

[8] Model Zoo, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md.

[9] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, arXiv: 1704.02767v1, 2018. https://arxiv.org/pdf/1804.02767.pdf.

[10] Yolo: Real-time object detection, https://pjreddie.com/darknet/yolo/.