DETECT ARTEFACTS OF VARIOUS SIZES ON THE RIGHT SCALE FOR EACH CLASS IN VIDEO ENDOSCOPY

Xiaokang Wang[1], *Chunqing Wang*[2]

[1]Department of Biomedical Engineering, University of California, Davis, Davis, CA 95616, USA
[2]Department of Ultrasound Imaging, Tiantan Hospital, Beijing, 100050, China

ABSTRACT

Detecting artefacts in video filmed in endoscopy is an important problem for downstream computer-assisted diagnosis. When tackling this problem, one challenge is that the size of an artefact varies in a wide range. The other challenge is that labeling endoscopic images is labor- extensive and is hard to outsource the labeling task to untrained people without the aid of doctors. In this report, we demonstrate how the performance of a Faster R-CNN model can be improved by scaling an image to the right scale before training and testing. The training method overcomes the issue that a convolution neural network trained on one scale barely works when detecting the same category of objects on a different scale. The method is totally independent of the model and can be easily adapted with other models. Besides, it saves time and memory by focusing on the patches that include objects when training the model. The source code* for this report will be made public upon the publishing of my solution.

1. INTRODUCTION

Endoscopy is a widely used clinical procedure for the early detection of numerous cancers (e.g., nasopharyngeal, gastric, colorectal cancers, bladder cancer etc.), therapeutic procedures and minimally invasive surgery. Video taken by the camera of an endoscope is usually heavily corrupted with multiple artefacts (e.g., pixel saturations, motion blur, defocus, specular reflections, bubbles, fluid, debris etc.). Accurate detection of artefacts is a core challenge in a wide range of endoscopic applications addressing multiple different disease areas. The importance of precise detection of these artefacts is essential for high-quality endoscopic frame restoration and crucial for realizing reliable computer assisted endoscopy tools for improved patient care.

In the last few years, convolution neural network (CNN) has outperformed previous non-CNN based methods in solving the object detection problem. The dominant CNN-based methods fall into two categories, one-stage approach and two-stage approach, with the former method shining in speed and the latter in accuracy. These two methods meet the demand in different fields. For example, in the field of self-driving car, speed is a prerequisite given an acceptable detection performance as a self-driving car has to react instantly. In the case of diagnosis in biomedical engineering, we can bear with slightly more computation time for higher accuracy. Since the advent of R-CNN [1], which is a two stage approach, this region-based detection method has become increasingly mature. Along the line, Fast R-CNN [2] introduced a RoI pooling operation that does forward pass on all the object proposals in an image simultaneously. Faster R-CNN [3] further speeds up R-CNN by training a region proposal network (PRN) using the feature maps generated by the convolution operations at the low level, without introducing much cost. Thus, I chose Faster R-CNN as the base framework in this challenge.

However, two challenges have to be solved when developing the model. One special challenge is that the size of the artefacts varies in a wide range and the other one is a limited number of labeled images (2,192 in total). The scale-related challenge is associated with the architecture of a CNN. The low level feature maps of a CNN capture features like edges and have a small receptive field, whereas the high-level features capture more semantic features, and have a larger receptive field [4, 5, 6]. Thus, the high-level features of small objects (e.g. less than 32 pixels) get mixed with features for background or objects nearby if the features do not disappear due to dimension reduction caused in feature extraction. e.g. For a feature stride of 32, the highest level features were shrunk 32 times compared to the raw image. For very large objects, the deeper layers suffer from extracting high-level semantic features due to failing to integrate low-level features given a limited feature stride.

To alleviate the problem caused by the wide range of object size, various solutions have been proposed. One category of solution focused on designing new CNN architectures to exploit the features at different levels. Under this paradigm, SSD [7] and MSCNN [8], use feature maps from different layers to detect objects at different scales. Although the features for small objects survive in the low-layer features, they lack semantic information which is supposed to be encoded in high level features. FPN [9], DSSD [10], STDN [11] integrate features at different layers. Another solution is to train a neural network on a multi-scale image pyramid, resulting

in a scale-invariant predictor [12]. Nevertheless, the previous solutions do not change the fact that high-level feature maps for small objects are mixed and the receptive fields for large objects are limited given an image and a CNN. Recently, a new training method that detects all objects at a proper scale by scaling up small objects and scaling down large objects has been reported in the state-of-the-art models, SNIPER [13] and TridentNet [14].

In this study, we demonstrated the successful application of the idea of detecting objects of various size at the right scale in detecting artefacts in endoscopy. The report is organized in such an order: datasets, methods, results, discussion and conclusion.

2. DATASETS

The training dataset consists of 2,192 endoscopic images (Fig. 1 A), in which seven categories of artefacts were labeled [15, 16]. The seven categories are pixel saturation, motion blur, specular reflections, bubbles, strong contrast, instrument and other artefacts. The size of an artefact varies in a range from a few pixels to one thousand pixels (Fig. 1 B). The number of objects in each category is from 453 to 5835 (Fig. 1 C). The performance of a model was tested on two datasets, one collected by the same endoscope and the other collected by a different endoscope to test the generalization ability of a model. The former and latter testing datasets comprise 195 and 51 images, respectively.

3. METHODS

The model we built is a Faster R-CNN with a FPN as the backbone. An FPN consists of mainly two parts, an encoder and an decoder, which is very similar to a U-Net [17] architecture developed for image segmentation tasks. Considering the memory capacity of our GPU (GTX 1070 16GB), We chose ResNet-50 as the workhorse of the encoder [18]. The implementation was based on a modularized implementation of mask R-CNN [19] in Pytorch [20]. The weights of the model were initialized with the weights trained on the COCO dataset except that the weights for the classification and regression head were initialized with random weights.

When training the model, we adapted the method proposed in [13] considering the class imbalance in our dataset and introduced data augmentation by strategically cutting a patch from an image for training. In specific, given all the bounding boxes (bboxes) in an image, k+1 bboxes were sampled from all the bboxes in this image (Fig. 2 A). k is the number of categories of objects in this image and 1 represents a random bbox. Such operation is to alleviate the class imbalance problem (Fig. 1 C) in our dataset. Then one bbox was sampled from the k+1 bboxes. Finally a patch of the image was cut out and scaled up or down depending on the size of the object in that patch.

When cutting a patch of the image (Fig. 2 B, step 1), the size of the patch and location of the patch was jittered, which allows us to generate not exactly the same patch every time even though the patch with the same object is selected. Note that the size of the patch is always larger than the object and a larger patch was cut if a smaller object exists in that patch. Otherwise, if the object were always in the center or same location in the patch, the model would not learn to detect objects but learn to localize objects assuming there is always an object, which is not true. The setting for the size of a patch (s_{patch}) is defined by this equation: $s_{patch} = r \times s_{bbox}$, where r = 4.5, 2, 1.5, and 1.2, respectively, for the cases, $s_{bbox} < 80$, 160, 350, and > 350. If no object exists in a random patch, a fixed size of patch was cut from an image.

After cutting a patch from an image, the patch was scaled up or down depending on the size of the object in that patch (Fig. 2 B, step 2). The scaling provides a zoomed-in view for small object objects and a zoomed-out view for large objects. Thus, both high-level and low-level feature maps will exist after an image passes a CNN. The scaling ratio (r) is inversely proportional to the size (s) of the object in the patch: r = 160/s and the size of all the objects are grouped into 6 bins. So the settings used here are (r=4, $s < 40$), (r=2, s¡80), (r=1, $s < 160$), (r=0.5, $s < 250$), (r=0.25, $s < 640$), and (r=0.13, $s > 640$). The reason for choosing such settings is that there will be 5 pixels in the last layer of the encoder if a raw input of 160 pixels is fed into the model, which has a feature stride of 32. A patch, which has no objects in it, was scaled with a random ratio on the fly.

After scaling up or down the patch cut from each image in a batch, objects that are too large/small were excluded. The thresholds for too small and too large objects are 32 and 2000, respectively. Choosing 32 as the threshold is because of the feature stride of the model is 32 and choosing 2000 is just because it is large enough.

In each training iteration, multiple patches from multiple images were cut and normalized as a batch by padding the patch with with the channel mean and concatenated, resulting in a batch of images, whose width and height are a multiple of 32 (Fig. 2 B, step 3). For a patch that is already a multiple of the stride size of the encoder, no padding was added. Since the padding is always on the bottom and right side if necessary, the coordinates of the bounding box does not change. Collating multiple samples and unifying them in size can be easily implemented in Pytorch[1]. For other details like how the bounding boxes were adjusted accordingly when cutting a patch from an image, see our code* on Github.

4. RESULTS

In inference, we tested one image on all the scales used in training (scales: 4, 2, 1, 0.5, 0.25 and 0.13). The coordi-

[1] https://pytorch.org

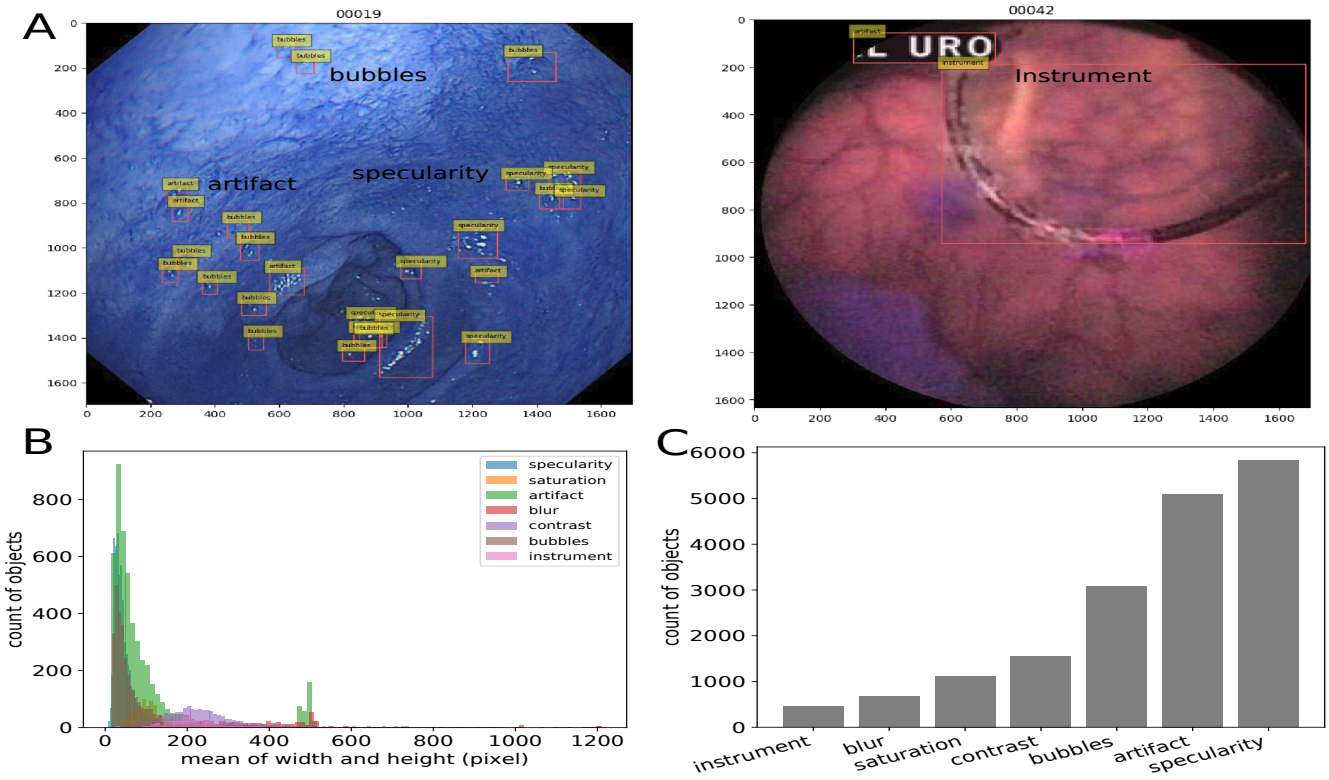

Fig. 1. **A**. Two sample endoscopic images. **B**. The distribution of the size of each category of artefacts in the training set. **C**. The distribution of the counts of seven categories of artefact in the training set.

nates of all the detected objects were transformed back to the original scale. To remove redundant bounding boxes, non-maximum suppression were conducted on all the detected bounding boxes for each category of object. It was observed that many false positive, which were small objects, were detected on the scale of 4, so I finally chose not to include the prediction on that scale.

The performance of our model was evaluated by a hybrid metric which was a weighted score of mean average precision (mAP) and Intersection over Union (IoU): 0.6 $\times$ mAP + 0.4 $\times$ IoU. We compared the performance of two ways for training the same model. One way is training the model on the whole image every time and the other is on patches generated following the method described above. The threshold for the probability when determining an object is 0.65. In the former case, the best overall score on the two testing sets was 0.221. For the latter case, we trained the network for 27,105 iterations (batch size is 8 in each iteration) and a significant boost in performance was observed. The score we reached was 0.293, which was among the top 10 teams on the leaderboard.

5. DISCUSSION & CONCLUSION

The training method boosted the performance of Faster RCNN in two ways. First, as we discussed in the Introduction section, it alleviates the scale variation problem by scaling up/down an object to the right scale to detect. Second, randomly cutting a patch which includes an object allows us to generate far more different training images compared to feeding the whole image to the model. Thus, cutting a patch serves as a data augmentation technique. Besides, it offers flexibility to deal with the class imbalance as we can choose which patch to cut from a training image, considering the distribution of the counts of all the classes.

Since the detected bboxes on all the scales were merged, a bbox was called if it was detected on any of the scales. Such an integration approach tends to report more false positive. One failure case we observed is that a false positive object does look like a true object because the model decides without considering the context of that object. We run into the case when an image is scaled up by 4 times. Thus the context information does matter and a context refinement probably corrects such kind of errors [21]. An alternative solution to solve this bias of this method can be feeding the detected bboxes as the input for the RoI pooling layer and merging the features generated by the model on an image pyramid. Since

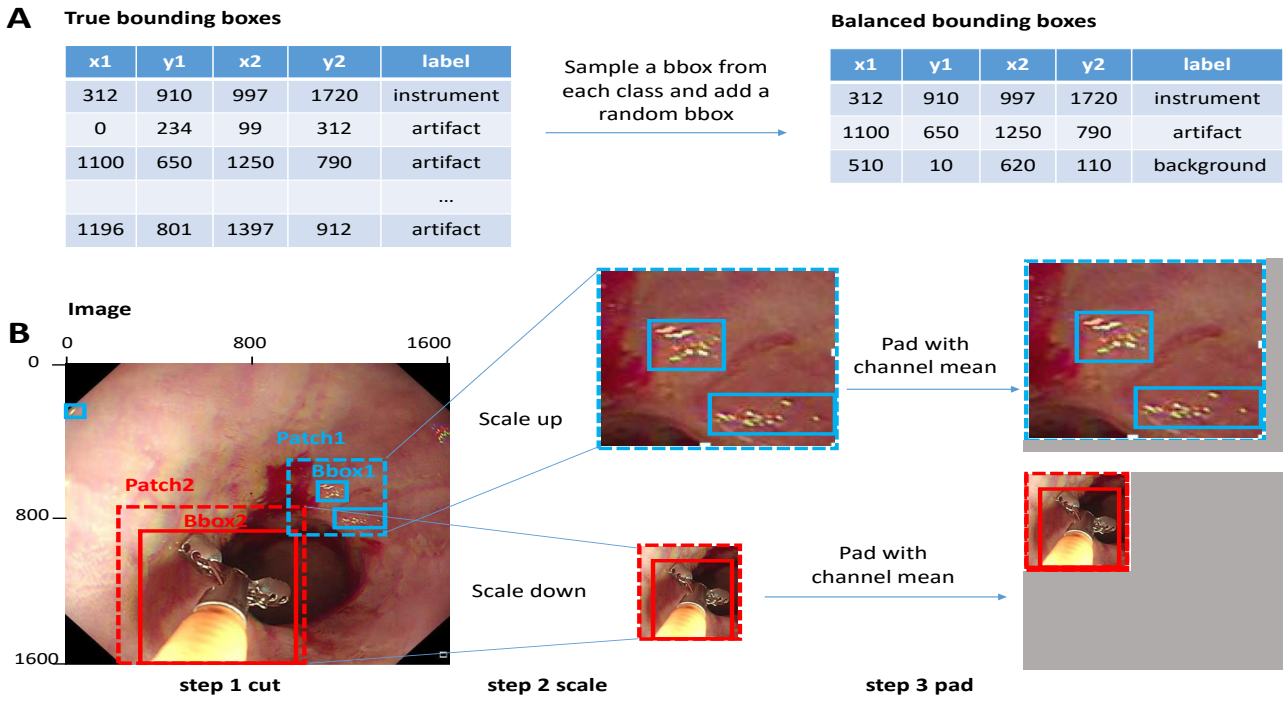

Fig. 2. The method for training the model to detect objects of various size on the right scale. **A**. a balanced set of bounding boxes was generated by sampling a bbox from each class and adding a random bbox. **B**. then a single bbox was sampled and a patch including the bbox was cut from the image. The patch was scaled down or up depending on the size of the object in the patch. Finally a batch of patches with unified size were generated.

the scaled down images include more context information, we expect the problem to be solved in this way.

To further boost the detection performance, the regular convolution operation in the FPN backbone can be replaced by deformable convolution operation will enhance the transformation modeling capacity of CNNs [22] or a newly proposed backbone designed for object detection task [**?**]. In conclusion, there is still room for improvement and we have demonstrated the performance of a Faster R-CNN model can be improved significantly by training and detecting the objects on the right scale.

6. REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[2] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[4] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[5] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[6] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3578–3587, 2018.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[8] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer, 2016.

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[10] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[11] Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 528–537, 2018.

[12] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[13] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9310–9320, 2018.

[14] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019.

[15] Sharib Ali, Felix Zhou, Christian Daul, Barbara Braden, Adam Bailey, Stefano Realdon, James East, Georges Wagnires, Victor Loschenov, Enrico Grisan, Walter Blondel, and Jens Rittscher. Endoscopy artifact detection (EAD 2019) challenge dataset. *CoRR*, abs/1905.03209, 2019.

[16] Sharib Ali, Felix Zhou, Adam Bailey, Barbara Braden, James East, Xin Lu, and Jens Rittscher. A deep learning framework for quality assessment and restoration in video endoscopy. *CoRR*, abs/1904.07073, 2019.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[20] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018.

[21] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[22] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.