ARTEFACT DETECTION IN VIDEO ENDOSCOPY USING RETINANET AND FOCAL LOSS FUNCTION

Ilkay Oksuz[1], *James R. Clough*[1], *Andrew P. King*[1*], *Julia A. Schnabel*[1*]

[1]School of Biomedical Engineering & Imaging Sciences , King's College London, UK

ABSTRACT

Endoscopic Artefact Detection (EAD) is a fundamental task for enabling the use of endoscopy images for diagnosis and treatment of diseases in multiple organs. Precise detection of specific artefacts such as pixel saturations, motion blur, specular reflections, bubbles and instruments is essential for high-quality frame restoration. This work describes our submission to the EAD 2019 challenge to detect bounding boxes for seven classes of artefacts in endoscopy videos. Our method is based on focal loss and Retina-net architecture with Resnet-152 backbone. We have generated a large derivative dataset by augmenting the original images with free-form deformations to prevent over-fitting. Our method reaches a mAP of 0.2719 and a IoU of 0.3456 for the detection task over all classes of artefact for 195 images. We report comparable performance for the generalization dataset reaching a mAP of 0.2974 and deviation from the detection dataset of 0.0859.

***Index Terms*—** Endoscopic artefact detection, focal loss, retina-net, class imbalance

1. INTRODUCTION

Endoscopy is a procedure in which the inside of the body is examined using a long, thin, flexible tube that has a light source and camera at one end, which allows visualization of the inside of organs on a screen. It is a widely used clinical procedure for the early detection of numerous cancers as well as for therapeutic procedures and minimally invasive surgery. A major handicap of endoscopy video frames is that they are subject to heavy corruption with multiple artefacts. The Endoscope Artefact Detection (EAD) challenge in the ISBI 2019 conference provides a multi-institutional dataset consisting of 7 different types of artefact (i.e. saturation, motion blur, specular reflections, bubbles, instrument, contrast and artifact). These artefacts not only cause difficulties in visualizing the underlying tissue during diagnosis but also affect any post-analysis methods required for follow-up (e.g. video mosaicking done for archival purposes and video-frame retrieval needed for reporting). 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 is crucial for realising reliable computer assisted endoscopy tools for improved patient care. An example ground truth bounding box annotations is visualized in Figure 1a.

Existing endoscopy workflows detect only one artefact class which is insufficient to obtain high-quality frame restoration as detailed in a comprehensive review about image quality estimation [1]. In general, the same video frame can be corrupted with multiple artefacts, e.g. motion blur, specular reflections, and low contrast can be present in the same frame. Furthermore, not all artefact types contaminate the frame equally. So, unless multiple artefacts present in the frame are known with their precise spatial location, clinically relevant frame restoration quality cannot be guaranteed. Another advantage of such detection is that frame quality assessments can be guided to minimise the number of frames that get discarded during automated video analysis.

2. RELATED WORKS

The existing works on endoscopic artefact detection are mainly focused on thresholding-based methods using the HSV [2] and RGB colour channels. Queiroz et al. [3] proposed to use a principal component analysis based detection algorithm of specular artefacts. Akbari et al. [4] proposed to use a non-linear SVM specular artefact detection using both HSV and RGB colour space information for segmentation of specular reflections. The SVM was trained with 12 statistical features including the mean and standard deviation of each channel of the RGB and HSV colour spaces.

The nature of this multi-class artefact detection challenge is in close relation to object detection challenges in computer vision (e.g. the COCO challenge [5]). The top performing algorithms on COCO and similar computer vision object detection challenges are based on convolutional neural network deep learning architectures. Current state-of-the-art object

This work was supported by an EPSRC programme Grant (EP/P001009/1) and the Wellcome EPSRC Centre for Medical Engineering at School of Biomedical Engineering and Imaging Sciences, Kings College London (WT 203148/Z/16/Z). We acknowledge financial support from the Department of Health via the NIHR comprehensive Biomedical Research Centre award to Guys & St Thomas NHS Foundation Trust with KCL and Kings College Hospital NHS Foundation Trust.

* Joint last authors.

detectors are based on a two-stage, proposal-driven mechanism. As popularized in the R-CNN framework [6], the first stage generates a sparse set of candidate object locations and the second stage classifies each candidate location as one of the foreground classes or as background using a convolutional neural network. Through a sequence of advances [7, 8], this two-stage framework consistently achieves top accuracy on the challenging COCO benchmark [5]. Despite the success of two-stage detectors, also one stage detectors are applied over a regular, dense sampling of object locations, scales, and aspect ratios. Recent work on one-stage detectors, such as YOLO [9], demonstrates promising results, yielding faster detectors with high accuracy. In this direction Lin et al. proposed RetinaNet [10], which is a one-stage object detector that matches the state-of-the-art COCO Avearage Precision (AP) of more complex two-stage detectors, such as the Feature Pyramid Network (FPN) [11] or variants of Faster R-CNN [7]. To achieve this result, class imbalance during training was identified as the main obstacle impeding one-stage detectors from achieving state-of-the-art accuracy and a new loss function that eliminates this barrier was proposed.

Class imbalance is one-key issue in the EAD 2019 multi-artefact detection challenge [12, 13], where the classes have an imbalanced distribution in the training set (e.g. specularity 43%, blur 3.5%, artifact 12%). Class imbalance is addressed in R-CNN-like detectors by two-stage cascade and sampling heuristics. The proposal stage rapidly narrows down the number of candidate object locations to a small number, filtering out most background samples. In this paper, we address this class imbalance by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples similar to [10]. Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. To evaluate the effectiveness of our loss, we design and train a simple dense detector based on RetinaNet. As highlighted in [10], when trained with the focal loss, RetinaNet is able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-art two-stage detectors. We use a loss function that acts as a more effective alternative to previous approaches for dealing with class imbalance. The loss function is a dynamically scaled cross entropy loss, where the scaling factor decays to zero as confidence in the correct class increases. Intuitively, this scaling factor can automatically down-weight the contribution of easy examples during training and rapidly focus the model on hard examples.

3. METHODS

RetinaNet is a single, unified network composed of a backbone network and two task-specific subnetworks. The backbone is responsible for computing a convolutional feature map over an entire input image and is an off-the-self convolutional network. The first subnet performs convolutional object classification on the backbone's output; the second subnet performs convolutional bounding box regression. The two subnetworks feature a simple design that we propose specifically for one-stage, dense detection. While there are many possible choices for the details of these components, most design parameters are not particularly sensitive to exact values as shown in the experiments. We detail the components of RetinaNet in the following sections.

3.1. Feature Pyramid Network Backbone

We adopt the Feature Pyramid Network (FPN) from [11] as the backbone network for RetinaNet. In brief, FPN augments a standard convolutional network with a top-down pathway and lateral connections so the network efficiently constructs a rich, multi-scale feature pyramid from a single resolution input image. Each level of the pyramid can be used for detecting objects at a different scale. FPN improves multi-scale predictions from fully convolutional networks (FCN), as well at two-stage detectors such as Fast R-CNN or Mask R-CNN. Following this, we build FPN on top of the ResNet architecture [14]. We construct a pyramid with levels P3 through P7, where l indicates pyramid level (Pl has resolution $2l$ lower than the input). As in [11] all pyramid levels have C = 256 channels. Details of the pyramid generally can be found in [11].

3.2. Anchors

We use translation-invariant anchor boxes similar to those in the orginal Retina-net [10]. The anchors have areas of 322 to 5122 on pyramid levels P3 to P7, respectively. As in [11], at each pyramid level we use anchors at three aspect ratios 1:2, 1:1, 2:1. For denser scale coverage than in, at each level we add anchors of sizes 20, 21/3, 22/3 of the original set of 3 aspect ratio anchors. This improve AP in our setting. In total there are A = 9 anchors per level and across levels they cover the scale range 32 - 813 pixels with respect to the networks input image. Each anchor is assigned a length K one-hot vector of classification targets, where K is the number of object classes, and a 4-vector of box regression targets. We use the assignment rule from RPN but modified for multi-class detection and with adjusted thresholds. Specifically, anchors are assigned to ground-truth object boxes using an intersection-over-union (IoU) threshold of 0.7; and to background if their IoU is in [0, 0.6). As each anchor is assigned to at most one object box, we set the corresponding entry in its length K label vector to 1 and all other entries to 0. If an anchor is unassigned, which may happen with overlap in [0.6, 0.7), it is ignored during training. Box regression targets are computed as the offset between each anchor and its assigned object box, or omitted if there is no assignment.

3.3. Classification Subnet

The classification subnet predicts the probability of object presence at each spatial position for each of the A anchors and K object classes. This subnet is a small FCN attached to each FPN level; parameters of this subnet are shared across all pyramid levels. Its design is simple. Taking an input feature map with C channels from a given pyramid level, the subnet applies four 3×3 convolutional layers, each with C filters and each followed by ReLU activations, followed by a 3×3 convolutional layer with KA filters. Finally sigmoid activations are attached to output the KA binary predictions per spatial location. We use C = 256 and $A = 9$ in most experiments. In contrast to RPN , our object classification subnet is deeper, uses only 3×3 convolutions, and does not share parameters with the box regression subnet. We found these higher-level design decisions to be more important than specific values of hyperparameters.

3.4. Box Regression Subnet

In parallel with the object classification subnet, we attach another small FCN to each pyramid level for the purpose of regressing the offset from each anchor box to a nearby ground-truth object, if one exists. The design of the box regression subnet is identical to the classification subnet except that it terminates in $4A$ linear outputs per spatial location. For each of the A anchors per spatial location, these 4 outputs predict the relative offset between the anchor and the ground-truth box. The object classification subnet and the box regression subnet, though sharing a common structure, use separate parameters.

3.5. Focal loss

Our novel Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. Formally, focal loss is a modified version the cross entropy loss, with tunable focusing γ parameter:

$$\text{Focal loss} = -\alpha_t(1 - p_t)^{\gamma} log(p_t)$$

where p_t is class-specific probability of belonging to a class and α is a weighting parameter.

There are two important properties of the focal loss, which makes it appealing for EAD 2019 challenge: (1) When an example is misclassified and p_t is small, the modulating factor is near 1 and the loss is unaffected. With increasing p_t, the factor goes to 0 and the loss for well-classified examples is downweighted. (2) The focusing parameter γ smoothly adjusts the rate at which easy examples are downweighted. When $\gamma = 0$, Focal loss is equivalent to CE, and as γ is increased the effect of the modulating factor is likewise increased. Intuitively, the modulating factor reduces the loss contribution from easy examples and extends the range in which an example receives low loss. The total loss is a combination between the focal loss and a regression loss on bounding boxes.

4. IMPLEMENTATION DETAILS

RetinaNet forms a single FCN comprised of a ResNet-FPN backbone, a classification subnet, and a box regression subnet. We use ResNet-152-FPN backbone to run our experiments. As such, inference involves simply forwarding an image through the network. To improve speed, we only decode box predictions from at most 200 top-scoring predictions per FPN level, after thresholding detector confidence at 0.36. The top predictions from all levels are merged and non-maximum suppression with a threshold of 0.5 is applied to yield the final detections. We trained our network using the Keras framework with Tensorflow library on nVidia P6000 GPU.

4.1. Focal Loss

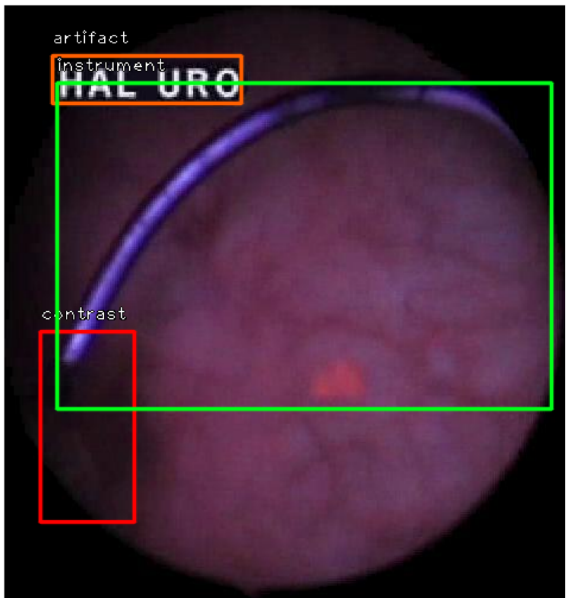
We use the focal loss introduced in this work as the loss on the output of the classification subnet. We find that $\gamma = 2$ and $\alpha = 0.25$ works well in practice and the RetinaNet is relatively robust. We emphasize that when training RetinaNet, the focal loss is applied to all 100k anchors in each sampled image. This stands in contrast to common practice of using heuristic sampling (RPN) or hard example mining (OHEM, SSD) to select a small set of anchors for each minibatch. The total focal loss of an image is computed as the sum of the focal loss over all 100k anchors, normalized by the number of anchors assigned to a ground-truth box. We perform the normalization by the number of assigned anchors, not total anchors, since the vast majority of anchors are easy negatives and receive negligible loss values under the focal loss. In general α should be decreased slightly as γ is increased, as highlighted in the original Retina-net paper.

4.2. Initialization

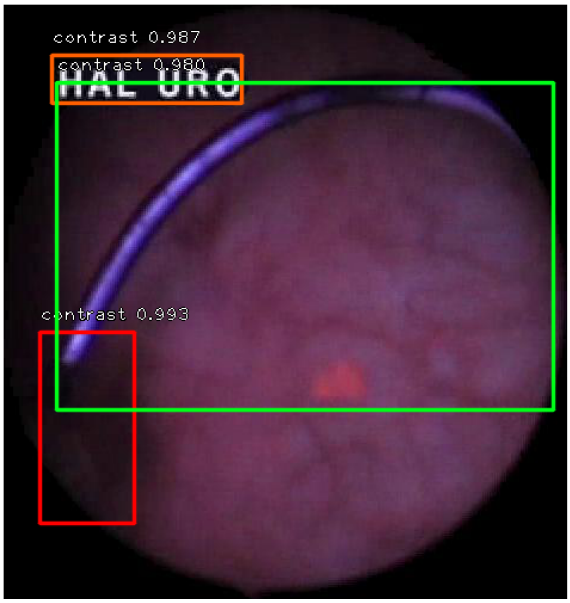
All new convolutional layers except the final one in the RetinaNet subnets are initialized with bias $b = 0$ and a Gaussian weight fill with $\sigma = 0.01$. For the final convolutional layer of the classification subnet, we set the bias initialization to b = log((1)/), where specifies that at the start of training every anchor should be labeled as foreground with confidence of . We use $\phi = .01$ in all experiments, although results are robust to the exact value. This initialization prevents the large number of background anchors from generating a large, destabilizing loss value in the first iteration of training.

4.3. Optimization

RetinaNet is trained with stochastic gradient descent (SGD). We use a minibatch of 3 size of 3 images. The model is trained for 10000 iterations with an initial learning rate of 0.001, which is then divided by 10 at 5000 and again at 750



(a) Example ground truth bounding boxes



(b) Predicted bounding boxes

Fig. 1: Example artefact detection and confidence scores from training detection set (result using 5-fold cross validation). The example was used in the validation set in this setup was not used during training of the network.

iterations. Weight decay of 0.0001 and momentum of 0.9 are used. The training loss is the sum the focal loss and the standard smooth L1 loss used for box regression. Training of the network took 26 hours.

4.4. Augmentation

Our scheme of image augmentations was designed to prevent overfitting to the set of training images, and so make our method more generalisable to the images in the test set. We assessed the effect of these augmentations by training the network with just the original training data and applying it to the test set images to produce artefact detections. In accordance to this is the observation that training without augmentations produces a much smaller final loss value as compared to training with augmentations. Having trained on the whole dataset for 10000 iterations with a batch size of 600 images, the final loss value without augmentations is 0.0082 but with augmentations is 0.0605. This clearly indicates significant over-fitting to the training dataset when augmentations are not used.

5. EXPERIMENTAL RESULTS

We used a stratified 5-fold cross validation strategy to optimize the parameters of the network. Table 1 summarizes the quantitative results achieved over 5-fold for each of the seven artefact classes. The data imbalance in between different classes and the how easily distinguishable each specific classes influences the specific mAP score for each class.

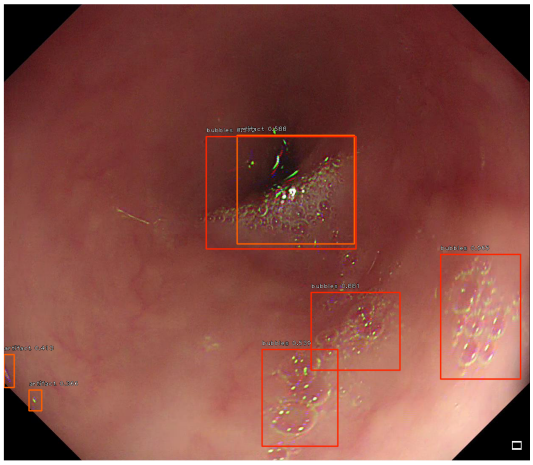


Fig. 2: Example artefact detections and confidence scores from test detection set.

We reach 0.2719 mean average precision (mAP) score on 195 cases over all 7 artefact classes. The intersection over union (IoU) for our predictions is 0.3456 for detection task over all classes. We report comparable performance for 51 images generalization dataset reaching a mAP of 0.2974 and deviation from detection dataset of 0.0859.

The visual result of 5-fold cross-validation for a case from validation cohort is visualized in Figure 1. The trained network is capable to generate bounding boxes with high confidence for an unseen case during training A qualitative result

Fold	specularity	saturation	artifact	blur	contrast	bubbles	instrument	mAP	IoU
0	0.5694	0.5908	0.6258	0.6053	0.6832	0.4962	0.6785	0.5245	0.4591
1	0.5619	0.5977	0.6518	0.6098	0.7037	0.5060	0.6858	0.5309	0.4562
2	0.5709	0.6193	0.6674	0.5969	0.7104	0.5112	0.6969	0.5401	0.4015
3	0.5613	0.6291	0.6689	0.6011	0.6974	0.5076	0.6897	0.5354	0.4209
4	0.5659	0.6072	0.6882	0.6185	0.7141	0.5213	0.6871	0.5442	0.4173
Mean	0.5659	0.6604	0.6063	0.70176	0.5085	0.9605	0.6876	0.53502	0.4310
D-Test	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.2719	0.3456
G-Test	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.2974	N/A

Table 1: 5-Fold validation average precision (AP) per class and intersection over union (IoU) results for seven classes. The AP results for each class, mean AP (mAP) and IoU results are reported for the validation over 5-fold (eg. for fold 5,0 the first fifth of images from the dataset were validation). The IoU column is the intersection-over-union difference between the bounding boxes inferred from the fold network and the ground truth bounding boxes. D-Test and G-test correspond to the detection and generalization test data for which per-class results are not reported in the challenge.

from detection test set is illustrated in Figure 2, with the prediction probabilities. The bubble and artefact classes are correctly identified in the example image. The ground truth is not available for this case.

6. DISCUSSION AND FUTURE WORK

In our experience it was clear when artefacts classes are poorly detected a significant factor is the size and total number of bounding boxes produced. The main difference in between different setups was dependent on the number of bounding boxes generated for artefacts in a neighbourhood. One critical factor in the final mAP score is the probability threshold used to include the detected artefacts. In future work, we aim to apply our algorithm on different artefact localization task for medical images (e.g. cardiac MR) with the availability of the training data.

7. REFERENCES

[1] Bernd Münzer, Klaus Schoeffmann, and Laszlo Böszörmenyi, "Content-based processing and analysis of endoscopic images and videos: A survey," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 1323–1362, 2018.

[2] Othmane Meslouhi, Mustapha Kardouchi, Hakim Allali, Taoufiq Gadi, and Yassir Benkaddour, "Automatic detection and inpainting of specular reflections for colposcopic images," *Open Computer Science*, vol. 1, no. 3, pp. 341–354, 2011.

[3] Fabiane Queiroz and Tsang Ing Ren, "Endoscopy image restoration: A study of the kernel estimation from specular highlights," *Digital Signal Processing*, 2019.

[4] Mojtaba Akbari, Majid Mohrekesh, Kayvan Najarian, Nader Karimi, Shadrokh Samavi, and SM Reza Soroushmehr, "Adaptive specular reflection detection and inpainting in colonoscopy video frames," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3134–3138.

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[6] 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*, 2014, pp. 580–587.

[7] 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*, 2015, pp. 91–99.

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

[9] Joseph Redmon and Ali Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[11] 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*, 2017, pp. 2117–2125.

[12] 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*, vol. abs/1905.03209, 2019.

[13] 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*, vol. abs/1904.07073, 2019.

[14] 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*, 2016, pp. 770–778.