EXPLORING DEEP LEARNING BASED APPROACHES FOR ENDOSCOPIC ARTEFACT DETECTION AND SEGMENTATION

Anand Subramanian, Koushik Srivatsan

Claritrics India Pvt Ltd, Chennai

ABSTRACT

The Endoscopic Artefact Detection challenge comprises tasks for detection and segmentation of artefacts found in endoscopic imaging, with a specific task for evaluating the generalization capacity of detection algorithms on external data. For the detection of artefacts, we train RetinaNet and Faster-RCNN models. To segment artefacts from the endoscopic images, we train a Deeplab v3 model and a U-Net model and also implement post-processing techniques such as the usage of an EAST text detector for detection of text artefacts and pixel-wise voting ensemble after applying test time augmentation. We observe that the RetinaNet model with a ResNet-101 feature extractor is the best performing model across all object detection tasks, while the U-Net performs best in the segmentation tasks. We also implement a model agnostic object tracking pipeline utilizing image correlation-based trackers to reduce the inference time of object detection models. We believe that this pipeline can enable real-time analysis of endoscopic images in systems with processing constraints.

1. INTRODUCTION

Endoscopy is an important clinical procedure that finds application in the diagnosis of medical ailments. However, the video footage captured through endoscopes may be riddled with artefacts, due to variations in contrast, blurs, among other artefacts. Therefore there is a requirement for algorithms that can detect, localize, and segment these artefacts. Detecting and localizing these artefacts would be of immense help in applying image restoration techniques to correct them, with the downstream benefits of reducing the impact of instrument errors in making medical diagnoses, ultimately improving endoscopic imaging. The motivation of the Endoscopic Artefact Detection and Segmentation challenge is to encourage research in this direction. Our work is an attempt towards tackling these problems by implementing algorithms specific to these tasks. Our main contributions in this work are as follows:

- A detailed analysis of the performance of RetinaNet with two ResNet [1] based feature extractors (ResNet-50 and ResNet-101) for the detection tasks, and the impact of test-time-augmentation.
- A detailed analysis of the performance of Deeplab-v3 model [2] and U-Net model [3] for the segmentation task along with the post-processing techniques applied.
- An implementation of a model agnostic tracking pipeline, using existing image correlation trackers for real-time inference of object detection models.

2. DATASETS

The artefact detection, sequence detection, and generalization tasks include specularity, bubbles, artefact, saturation, contrast, blood, blur, and instrument as classes to be detected. The training set for this challenge [4, 5, 6] was released in phases, initially as a set of 2200 images with bounding boxes annotations. Sequence data, taken from videos, was provided as a total of 232 images in the next phase, with a set of 99 images provided in the final phase. The artefact segmentation dataset included instrument, specularity, artefact, bubbles, and saturation as artefact classes. The training data for this comprised 474 images and their corresponding class wise segmentation masks, with the other data releases being the same data as released for detection, with masks for only these five classes.

3. METHODS

3.1. Object detection

In this work, we train two RetinaNet models with ResNet-50 and ResNet-101 feature extractors and a Faster-RCNN [7] model with an Inception v2 [8] feature extractor, for the detection task. We use a Keras implementation of RetinaNet [9] and the Tensorflow implementation of Faster-RCNN [10] for training the models. We apply on the fly augmentation techniques such as rotation, shear, random-image-flip, image contrast, brightness, saturation, and hue variations randomly

during training, to improve generalization of the object detection algorithm. During training, when augmentation is applied to the images on the fly, the model is not only exposed to the actual raw image, but also to the transformed versions of the data across iterations. The concept of Test Time augmentation builds on this by providing augmented test images to the model, in the assumption that the model would output better predictions since it has learned features from images which have had the same transformations applied while training. This technique is implemented at inference using an ensemble framework [11], and the final output bounding boxes are ensembled based on their Intersection-over-Union values with a majority criterion. The augmentations applied to the image at inference are horizontal flip and sharpening. Both the RetinaNet models predict outputs for the augmented image and for the image without augmentation. The outputs from each model are then ensembled to get the final prediction.

3.2. Sequence Detection

The other sub-task involved in this challenge is the detection of image sequences. We observe that the problem of sequence detection is primarily a video object detection problem. To this end, we implement an object tracking pipeline using the *Dlib* Correlation Tracker [12, 13] and the Discriminative Correlation Filter-with Channel and Spatial Reliability (CSRT) tracker [14] in tandem with an object detection model. We design this as a baseline model agnostic pipeline, working with any algorithm/model that outputs frame-wise bounding boxes, scores, and classes. Some of the problems with using object detection directly for videos are the latency issues involving the model, as detection has to be made once every frame. This overhead may be unacceptable for deep models with high inference times, especially in systems with processing constraints. Other issues may involve identifying and associating one object across multiple frames. This contribution is intended to explore building systems that can function even in resource-constrained setups, by reducing model latency, where model predictions are required for only every N frames, instead of every single frame. We design the object tracking pipeline, such that the bounding boxes from one frame are tracked across multiple frames, with minimal drops in accuracy whilst being robust to movement artefacts, and decreasing the overall processing time, as the model does not have to predict over all the frames. We control the rate at which the model needs to refresh the trackers with a parameter called the Window Size (W), which is the number of frames after which the model re-initializes the trackers. This is required for both the correlation trackers.

3.3. Semantic Segmentation

For artefact segmentation, we train a Deeplab v3 model with an Xception backbone [15], as well as a U-Net model with a

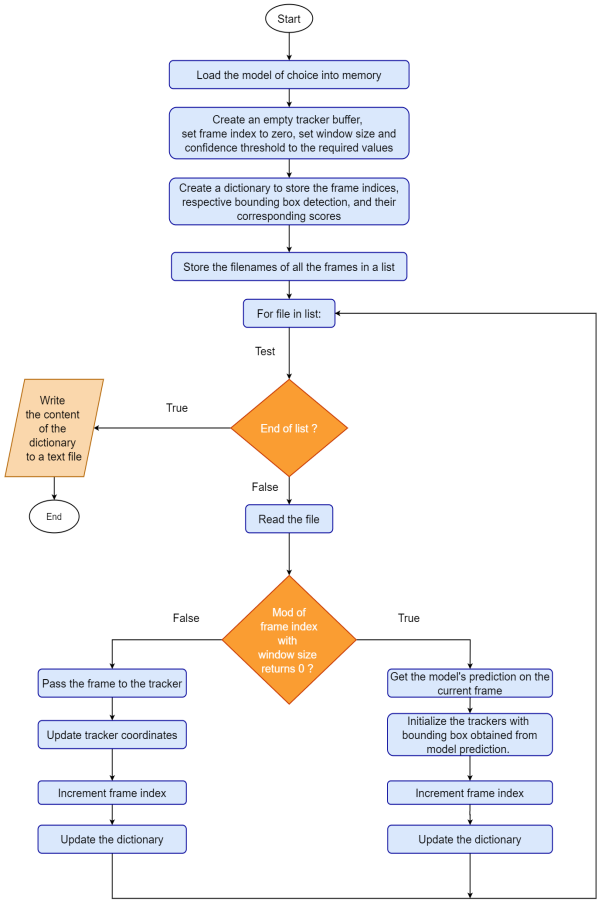

Fig. 1: Flowchart of our tracker pipeline

ResNet-50 backbone pre-trained on Imagenet [16]. We treat this problem as a multi-label segmentation problem due to multiple overlapping masks. We use Keras implementations of Deeplab [17] and U-Net [18] for this task. On analysis of the images, we find text segments in images labeled as artefacts and use a pre-trained EAST [19] text detector to capture these regions. This is done after observing that the models were unable to capture the text parts accurately during segmentation. For the purpose of image augmentation, we use scale variation, random flips, rotation, the addition of noise, cropping, blurring, hue, saturation, contrast changes, and sharpening. These augmentation techniques are randomly applied on the fly during training, with only one transformation applied for a batch of images. We perform Test-Time-Augmentation for the Deeplab and U-Net models, and ensemble their outputs using pixel-wise voting (majority), and finally, add the text masks captured using the EAST Detector to the result. We implement a custom pipeline that ensembles the two segmentation models using Test-Time-Augmentation and integrates the EAST text detector. We use the *imgaug* library [20] for building train-time and test-time augmentation pipelines.

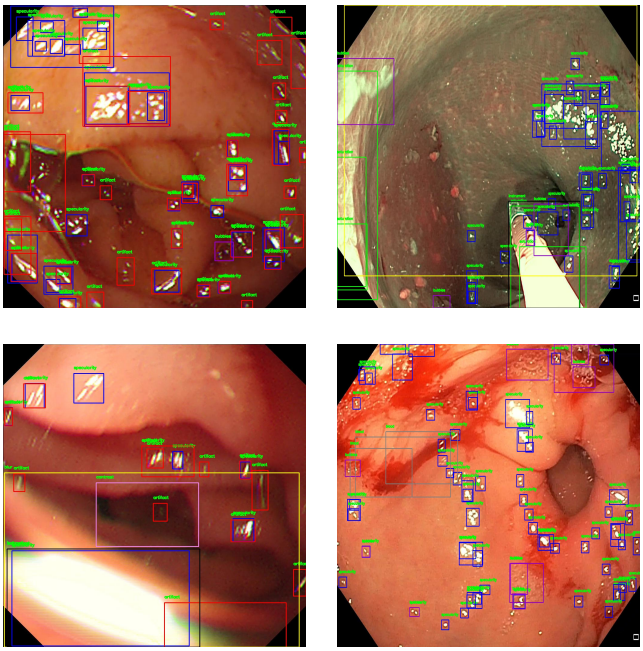

Fig. 2: Object detection results of RetinaNet (R-101) on the final test data. The box color representation for each class are as follows. Artefact , Bubbles, Saturation, Instruments, Contrast, Blur, Specularity and Blood.

Method	mAP	IoU
RetinaNet (R-50)	0.15754	0.24115
RetinaNet (R-50) w TTA	0.13515	0.29334
Faster-RCNN	0.06484	0.12803

Table 1: Object detection results on partially released detection data

4. RESULTS

In the tabulation of results, the RetinaNet with a ResNet-50 feature extractor is mentioned as RetinaNet (R-50) and the RetinaNet with ResNet-101 feature extractor as RetinaNet (R-101). The use of Test-Time-Augmentation is abbreviated as TTA.

4.1. Results on first phase of test data

From the results on the partial test data for object detection provided in Table 1, we find that the usage of test-time augmentation for the RetinaNet model increases the overall IoU by almost 4 percent, but the overall mAP is reduced by 2 percent compared to the original model without applying this method. The Faster-RCNN, however, performs poorly in comparison with the RetinaNet model, in terms of both the metrics.

Method	mAP	IoU	FPS
RetinaNet (R-50)	0.20061	0.25118	0.96667
RetinaNet (R-50) w *Dlib* tracker	0.15386	Nil	3.68767
RetinaNet (R-50) w CSRT tracker	0.14286	Nil	2.30704

Table 2: Sequence detection results on partially released data

Method	mAP	deviation
RetinaNet (R-50)	0.20713	0.12230
Faster-RCNN	0.10823	0.09380

Table 3: Object detection results on partially released generalization data

Method	Segmentation score	DSC score
Deeplab+Unet+East Ensemble	0.49666	0.42946
Deeplab	0.45742	0.39998

Table 4: Artefact segmentation results on partially released segmentation data

Object detection method	Sequence detection method	mAP	deviation
RetinaNet (R-101) (w/o) TTA	RetinaNet (R-101) (w/o) TTA	0.2151	0.0762
RetinaNet (R-101 and R-50) (w) TTA	RetinaNet (R-101) + CSRT tracker	0.1537	0.0419
RetinaNet (R-101 and R-50) (w) TTA	RetinaNet (R-101) + *Dlib* tracker	0.1502	0.0419

Table 5: Object detection results on the detection and sequence data

Table 2 contains the results of the Retinanet (R-50) model and the trackers used in tandem with it for the sequence detection task on the partial data. The methods are also benchmarked in terms of frames-per-second.

Table 3 contains the results of the models on partially released generalization data. We observe that the RetinaNet performs better overall compared to the Faster-RCNN across both metrics. This could in part be due to the better generalization of the RetinaNet as a result of applying augmentation while training, whilst augmentations were not applied while training the Faster-RCNN model. Table 4 contains the results of the model on partially released segmentation test data. We find that the ensemble of the Deeplab, the U-Net and the EAST Detector has a significantly higher segmentation score and DSC score, compared to the Deeplab model in isolation. Here, the U-Net is not specifically benchmarked.

4.2. Results on full test data

Table 5 contains the results of the models on the complete detection and sequence data. For the results on the final test data, only the combined mAP scores of the detection and se-

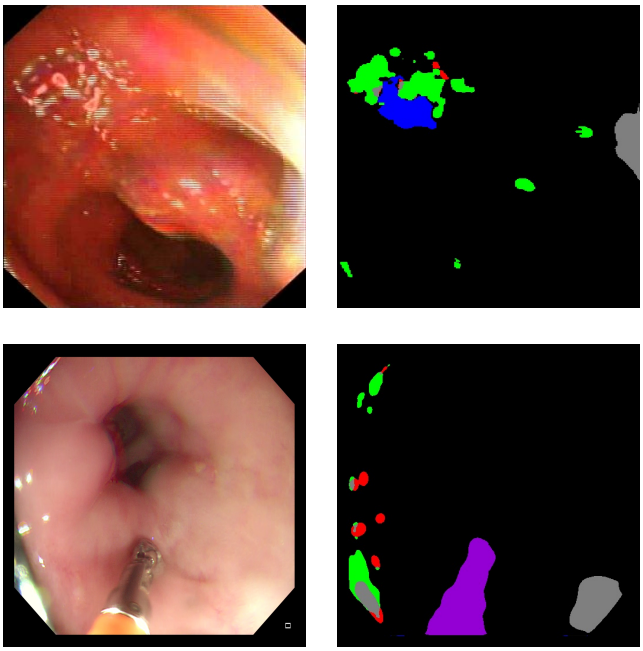

Fig. 3: Semantic Segmentation Results of U-Net on the final test data. The color representation for each class are as follows. Instrument, Specularity, Artefact, Bubbles and Saturation.

Method	FPS
RetinaNet (R-101) + *Dlib* tracker	2.57412083
RetinaNet (R-101) + CSRT tracker	1.19047612
RetinaNet (R-101) w/o tracker	0.68269294

Table 6: Frames per second results of the tracking and detection systems

quence task was provided whilst other metrics were not provided. There is some ambiguity in analyzing the exact impact of the techniques. In one submission, we apply the RetinaNet with ResNet-101 feature extractor for both tasks. In the other two submissions, we apply the ensemble of RetinaNet models separately for the detection tasks, and the object tracking solutions for the sequence tasks. We find that our RetinaNet model with ResNet-101 feature extractor has a higher mAP score, compared to the results of ensembling RetinaNet with ResNet-50 and ResNet-101 feature extractors using test-time augmentation.

We observe that the complete sequence detection test data was released as one folder comprising frames belonging to two different videos. Since our tracking pipeline takes frames belonging to one video as input for processing, we group the sequence data frames based on the video they belonged to and run inference individually for each batch of frames. We benchmark the frames-per-second performance of the

Method	mAP	deviation
RetinaNet (R-50) w/o TTA	0.2121	0.2188
RetinaNet (R-101) w/o TTA	0.2020	0.1920
RetinaNet (R-101 and R-50) w TTA	0.1481	0.1250

Table 7: Object detection results on generalization data

Method	Segmentation Score	deviation
Unet	0.5012	0.2648
Deeplab+Unet+East Ensemble	0.4863	0.2751
Deeplab	0.4314	0.2985

Table 8: Artefact segmentation results on full test data

RetinaNet-101 with both the trackers, similar to the one carried out in Table 2. We benchmark the performance on an Intel(R) Core(TM) i7-8750H CPU with 6 CPU cores by averaging the frames-per-second across five runs on the sequence data. A window size parameter of 5 was set for running the tracker pipeline. The results are shown in Table 6. We also carry out the inference of the RetinaNet model on the CPU. We find that the *Dlib* and CSRT trackers are significantly faster than just the RetinaNet model. This can be attributed to two reasons:

1. We take the model's inferences once every five frames for the tracker pipelines as compared to per frame for the RetinaNet model without the tracker.

2. The tracking process carried out across the frames is faster than the model's inference time per frame. This validates our assertion that a tracking system can be more reliable for building systems on endoscopic analysis where memory and processing resources may pose constraints.

Table 7 contains the results of the models on the fully released test data for the generalization task. We observe that the RetinaNet (R-50) without test time augmentation performed the best in the generalization task compared to the other two results.

Table 8 contains the results of the Deeplab model, the U-Net Model, and the ensembled model for the segmentation task on the full test data. For the final test data, only the segmentation score is provided, without the additional metrics as provided for the partial release data. The U-Net model is benchmarked only for the full test data. While the ensemble model is a considerable improvement over the results of the Deeplab model, the U-Net model outperforms the others on the segmentation score.

However, a key area where our ensemble model is limited is in the segmentation of instruments from endoscopic images.

As observed from Fig. 4, the ensembled model is unable to pick out metal bands linking the edges of the instruments.

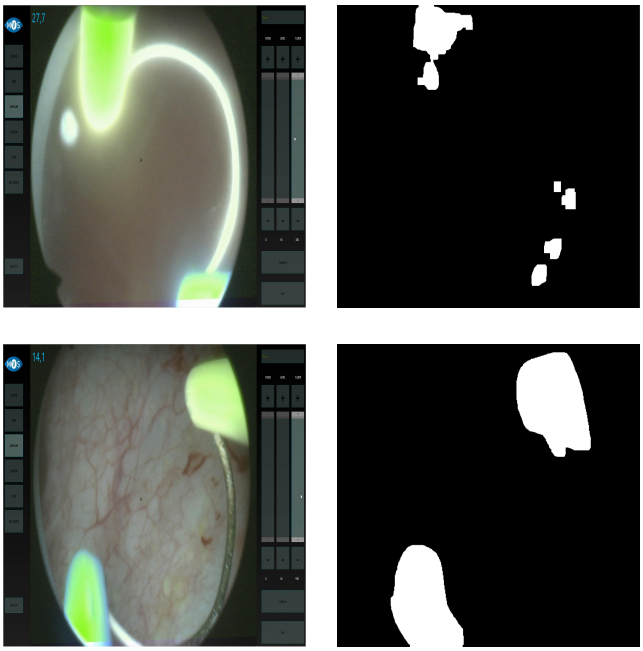

Fig. 4: Images of instruments and mask predicted by the ensemble model

This is observed across multiple instances and is a limitation in the predictions that are output by the model. Methods to tackle these could include the usage of image processing techniques involving region growth to link the edges.

5. DISCUSSION & CONCLUSION

We present the results of the models trained for the purpose of detecting, localizing and segmenting endoscopic artefacts. We train RetinaNet and Faster-RCNN models to detect and locate endoscopic artefacts, and Deeplab v3 and U-Net models for the purpose of segmenting endoscopic artefacts. We implement a model agnostic object tracking pipeline for the sequence detection task, utilizing image correlation-based trackers, and observe its impact on model inference time. We also implement post-processing techniques, such as test-time-augmentation, ensembling, and text detection, and present a study of their impact on the released test data. Further work could concentrate on improving the segmentation of instruments from images, as well as exploring the performance of different models for segmentation in general.

6. ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards the research and development team at Claritrics India for their help and guidance.

7. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241, 2015.

[4] Sharib Ali, Felix Zhou, Christian Daul, Barbara Braden, Adam Bailey, Stefano Realdon, James East, Georges Wagnieres, Victor Loschenov, Enrico Grisan, et al. Endoscopy artifact detection (ead 2019) challenge dataset. *arXiv preprint arXiv:1905.03209*, 2019.

[5] 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. *arXiv preprint arXiv:1904.07073*, 2019.

[6] Sharib Ali, Felix Zhou, Barbara Braden, Adam Bailey, Suhui Yang, Guanju Cheng, Pengyi Zhang, Xiaoqiong Li, Maxime Kayser, Roger D. Soberanis-Mukul, Shadi Albarqouni, Xiaokang Wang, Chunqing Wang, Seiryo Watanabe, Ilkay Oksuz, Qingtian Ning, Shufan Yang, Mohammad Azam Khan, Xiaohong W. Gao, Stefano Realdon, Maxim Loshchenov, Julia A. Schnabel, James E. East, Geroges Wagnieres, Victor B. Loschenov, Enrico Grisan, Christian Daul, Walter Blondel, and Jens Rittscher. An objective comparison of detection and segmentation algorithms for artefacts in clinical endoscopy. *Scientific Reports*, 10, 2020.

[7] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[9] keras-retinanet. https://github.com/fizyr/keras-retinanet.

[10] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and

Others. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[11] A. Casado-Garca and J. Heras. Ensemble methods for object detection. 2019. https://github.com/ancasag/ensembleObjectDetection.

[12] Martin Danelljan, Gustav Hger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. pages 65.1–65.11, 01 2014.

[13] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[14] Alan Lukei, Tom Voj, Luka ehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. *International Journal of Computer Vision*, 126, 11 2016.

[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] bonlime. keras-deeplab-v3-plus. *GitHub repository*. https://github.com/bonlime/keras-deeplab-v3-plus.

[18] Pavel Yakubovskiy. Segmentation models, 2019.

[19] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. *CoRR*, abs/1704.03155, 2017.

[20] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gbor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernndez, Franois-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020.