CEUR-WS.org/Vol-2886/paper10.pdf

Detection Of Polyps During Colonoscopy Procedure Using YOLOv5 Network

Tianyuan Gan[a], Zhenzhou Zha[a], Chunyong Hu[a] and Ziyi Jin[a]

[a] *Biosensor National Special Laboratory, Key Laboratory of Biomedical Engineering of Ministry of Education, College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310027, P. R. China.*

Abstract

Although several methods for detecting and segmenting polyps during colonoscopy procedures have been established, their generalization abilities have yet to be assessed due to a lack of a unified benchmark on high-quality datasets. The 3rd International Endoscopy Computer Vision Challenge and Workshop (EndoCV2021) offers a detailed, well-curated, and well-defined colonic polyp dataset that includes multi-center data with different polyp sizes and imaging modalities. It aims to determine the generalizability of each participant's polyp detection and segmentation process. We present our method for detecting polyps in this paper. On this job, we compare three state-of-the-art object detection baselines: EfficientDet, ScaledYOLOv4 and YOLOv5, and pick the best one (YOLOv5). To boost the performance even more, we apply several data augmentation methods, hyperparameter evolution, multi-scale training, model ensemble, and test-time augmentation to the baseline. During the test step, we also use the existing knowledge from colonoscopy to refine our post-processing parameters. Finally, on the round-I test, our system achieves a detection score of 0.7948, and on the round-II test, it achieves a detection score of 0.8824.

Keywords

Object detection, colonoscopy, polyps

1. Introduction

Deep learning-based computer vision (CV) is one of the most prominent research fields nowadays. As a consequence, in minimally invasive surgery, which is primarily focused on endoscopes, an increasing range of CV applications can be observed. The endoscopic vision system was created to aid in endoscopic surgery, especially in gastroenterology [1, 2, 3]. How to identify and segment polyps easily and reliably during colonoscopy has also become a key research concern in the new computer-aided diagnostic (CAD) framework. Although many methods have been developed to perform polyp detection and segmentation during colonoscopy, the generalization ability cannot be evaluated due to the lack of a unified benchmark on high-quality data sets. The 3rd International Endoscopy Computer Vision Challenge and Workshop (EndoCV2021) [4] provides a comprehensive, well-curated, and defined colonic polyp dataset that collects data from multiple centers with different polyp sizes and imaging modalities. The previous challenge EndoCV2020 [5] and EAD2019 [6] also focused on the detection and segmentation in endoscopic images.

3rd International Workshop and Challenge on Computer Vision in Endoscopy (EndoCV2021) in conjunction with the 18th IEEE International Symposium on Biomedical Imaging ISBI2021, April 13th, 2021, Nice, France

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

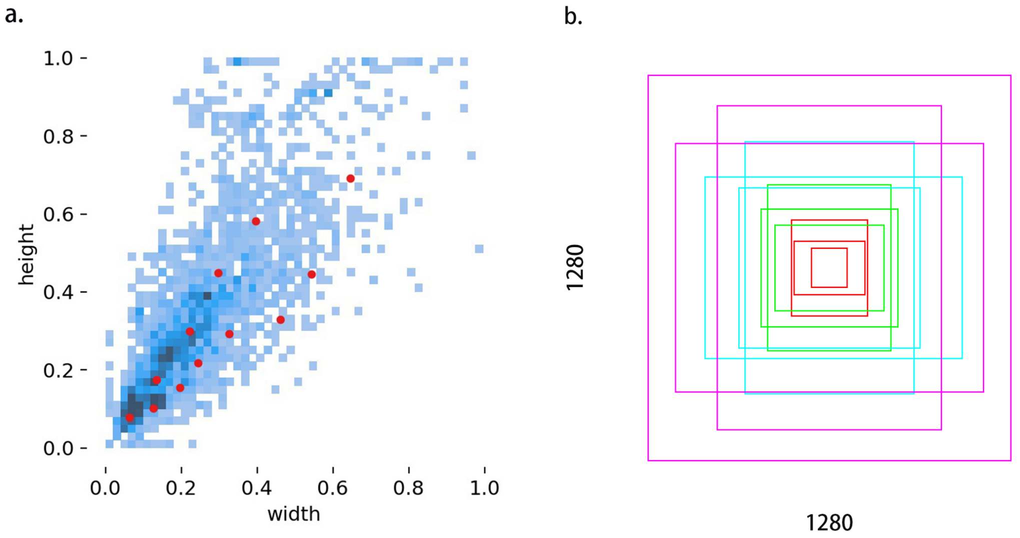

Figure 1: Analysis result of bounding boxes and the design of anchors. a. The red points stand for the cluster centers of the size of bounding boxes with K-means algorithm. b. The multi-scale anchors for YOLOv5 were designed based on the cluster centers of the size of bounding boxes.

In this article, we will use various architectures including EfficientDet [7], ScaledYOLOv4 [8] and YOLOv5 [9], and compare their performance to apply and train the state-of-the-art deep learning models for the detection task.

2. DATASETS AND DATA ANALYSIS

EndoCV2021 offers a dataset for polyp detection and segmentation, which contains 1449 single frame endoscopic images from five centers and 3601 sequence endoscopic images [4]. In total, the dataset contains 2713 positive frames with bounding boxes from professional physicians for polyps and 2337 negative frames. In the sequence data, 10 negative-only sequences of a total of 1808 frames are provided. However, pure negative frames are not necessary for the detection task, because the feature of negative samples can be obtained from the background area outside the annotations area of positive samples. In order to avoid the problem of sample imbalance, the negative frames from the sequence data are not used in our train dataset. Although using data balancing method such as weighted loss can also solve this problem, it will increase the time consuming of training procedure. And it won't significantly improve the performance of the model because the areas of background in positive frame are already large enough. However, the negative frames from the single frame data are preserved for improving the generalization ability of the model.

The images from the five centers are quite different in size and color. To ensure the balance of the training set, 10% samples of each center data and positive sequence data are chosen as the validation set. And the remaining samples are chosen as the training set. Finally, we have a training set of 2910 frames and a validation set of 332 frames.

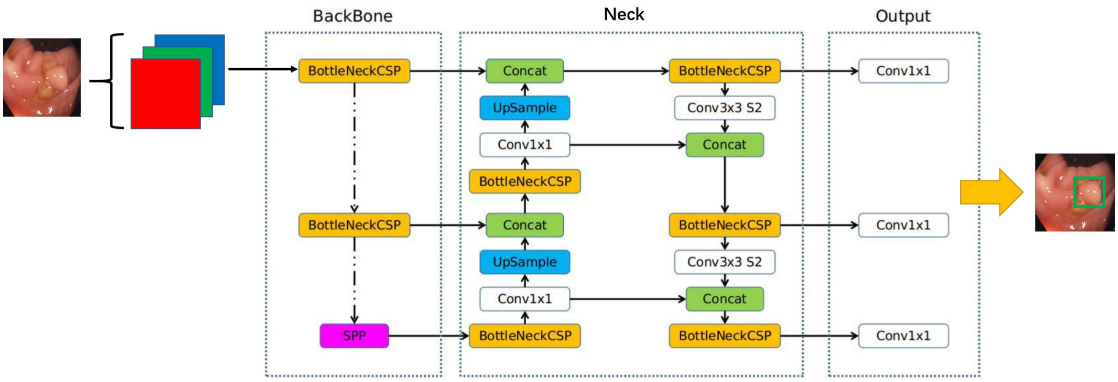

Figure 2: The architecture of YOLOv5

Besides, for designing the hyperparameters of the detection network, we analyze the dataset. The image sizes of the samples cover from 384×288 to 1920×1080. The average size of the images is 1486×1019. At the same time, K-means algorithm is used to analyze the size of the bounding boxes for the design of the anchor. The analysis results of the bounding boxes and the designed anchors are shown in Fig 1.

3. METHODS

In this section, we will introduce the best model we developed for the test dataset. We implemented our polyp detection method for EndoCV2021 using YOLOv5 as our baseline. YOLOv5 is a new state-of-the-art single-stage object detection algorithm. It achieved the nearly top mAP in the COCO dataset [10] as well as inference time unreachable for other baselines. The models were trained on two Tesla V100 NVIDIA GPUs. Some common python libraries including opencv, pillow, pytorch, torchvision, tqdm, numpy were used to build the pipeline of YOLOv5.

3.1. Pre-processing

Although our task has only one class of object, selection bias and sample imbalance still exist because the training dataset consists of multicenter data and the images vary from centers in their style and amount. On the other hand, the original training set only has 2910 images, it is easy to cause the model overfitting. To enhance the generalization of the model, all images of training dataset were randomly augmented every epoch before loading into the model using hue adjustment, saturation adjustment, value adjustment, rotating, translation, scaling, shearing, perspective, up-down flipping, left-right flipping, mosaic and mixup. Among the above data augmentation methods, mosaic and mixup are two most efficient ways. Fig 3 visualizes an example of these two methods. The first method stitches four pictures and the corresponding bounding boxes together in the form of 2×2 to improve the performance of small object detection. The second method multiplies the two pictures by different coefficients and stacks them together could significantly reduce the overfitting.

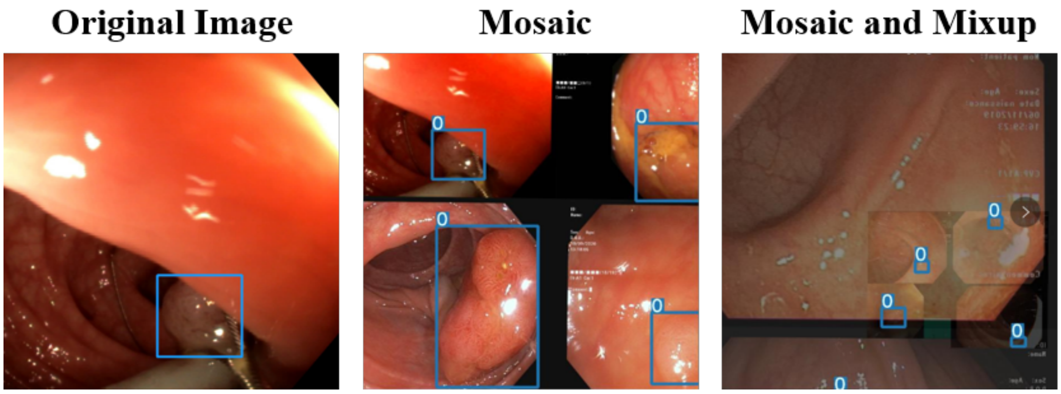

Figure 3: Two efficient data augmentation methods

3.2. Training procedure

We chose the yolov5x6 model as our baseline. The ‘x’ means it is the model with deepest and widest backbone in YOLOv5 series, which has the best feature extraction ability. The ‘6’ means ‘P6’, which adds a larger object output layer P6 to adapt the inputs with higher resolution (yolov5x for 640×640 inputs and yolov5x6 for 1280×1280 inputs, the average input size of our training dataset was 1486×1019). Besides, we also used the K-means clustering algorithm to calculate the sizes of the anchors. At the model initial stage, we used the transfer learning strategy which pre-trained the weights on the COCO dataset for 300 epochs. The model was then fine-tuned with an SGD optimizer until it stopped through early stopping to prevent overfitting, resulting in additional 90 epochs. Hyperparameters of the training procedure such as initial learning rate, final cycle learning rate, SGD momentum, optimizer weight decay, number of warmup epochs, box and class loss gain and the probability of various data augmentation could significantly influence the final performance of the model. Therefore, we refined the hyperparameters through hyperparameter evolvement based on genetic algorithm. This procedure usually requires hundreds or thousands of GPU hours. Thus, we used the hyperparameter which has evolved on COCO dataset for 306 generations and get a relative better performance in training object detection models because the limited challenge time. Furthermore, we used multi-scale training to train our models. The size of the input image has a significant impact on the performance of the detection model. Multi-scale is one of the most obvious tricks to improve the model generalizability. A feature map that is dozens of times smaller than the original image is generated in the backbone of the model, which makes it difficult for the features of small objects to be captured by the feature extraction network. By inputting images with various sizes either larger or smaller for training, the robustness of the detection model to different sizes of objects can be improved to a certain extent. Therefore, we change the size of the input images to a random value between 640 to 1920 (+/- 50% scaling for 1280) every epoch when dataloader load the images into the model at the beginning of the epoch.

3.3. Post-processing

We use non-maximum suppression (NMS). In the selection of the IoU threshold, we applied prior knowledge from the colonoscopy domain and the training dataset. We noticed that the number of polyps per frame of colonoscopy is usually very small and the distribution is relatively sparse unless it is a polyposis patient. Besides, by analyzing the images of the training set, we find that the annotators tend to put the polyps into the same bounding box if the polyps are close to each other. Therefore, we set the IoU threshold to 0.3, which is a relatively small value. By selecting this small IoU threshold, the bounding boxes of the neighboring polyps which has a big IoU value will be suppressed by the NMS algorithm and remain only one final box.

3.4. Inference procedure

When inferring on the test set, we used Test Time Augmentation (TTA). It will create multiple different versions of the input image, including cropping of different regions and changing the zoom level and so on, and load them into the model. Then the average output of multiple versions will be calculated as the final output of the image. To fully exert the effect of TTA, we increased the image size of the test stage to 1.2 times the size of the train stage (1280×1280 for training; 1536×1536 for testing). Another trick we used in this stage is model ensemble. We used 20 models with different weights generated during the training stage. All weights were produced between model convergence and model overfitting and achieved a similar and relatively optimal performance in our validation set.

4. RESULTS

Table 1 compares the results of the three SOTA object detection baselines on our validation dataset of the EndoCV2021 detection task. To make a fair comparison, we unified the input size of the images to 1280×1280, and chose the corresponding model (YOLOv5x-P6, ScaledYOLOv4-P6, EfficientDet-D6). The results show the YOLOv5 obtained the best mAP:0.5:0.95 of 0.5635. Therefore, we chose YOLOv5x-P6 as our final baseline. The results of the ablation experiment on the same validation set are provided in Table 2. Our best single model got a mAP:0.5:0.95 of 0.6843 on the validation set. With ensemble 20 different models generated during the training process, we get the best performance of 0.7178 mAP:0.5:0.95. Table 3 summarizes the detection results of our submission for the 2 test sets. For round-I test dataset, our best detection performance was achieved by 7 ensembled models, which got a score$_d$ of 0.7948 ± 0.8375 with the speed of 0.725 seconds per image (s/img). For round-II, the best score$_d$ of 0.8824 ± 0.2743 was obtained by ensemble 20 models with a IoU-thres of 0.3. The speed of this method is 2.056s/img. Besides, we also tested the performance of a single model without test-time augmentation (TTA) on the test set round II. The model showed a score$_d$ of 0.7696 ± 0.2331 with a speed of 0.035s/img, which has the potential to be applied to real-time colonoscopy in the clinical practice. Fig 4 visualizes the difference between the bbox of our prediction and ground truth.

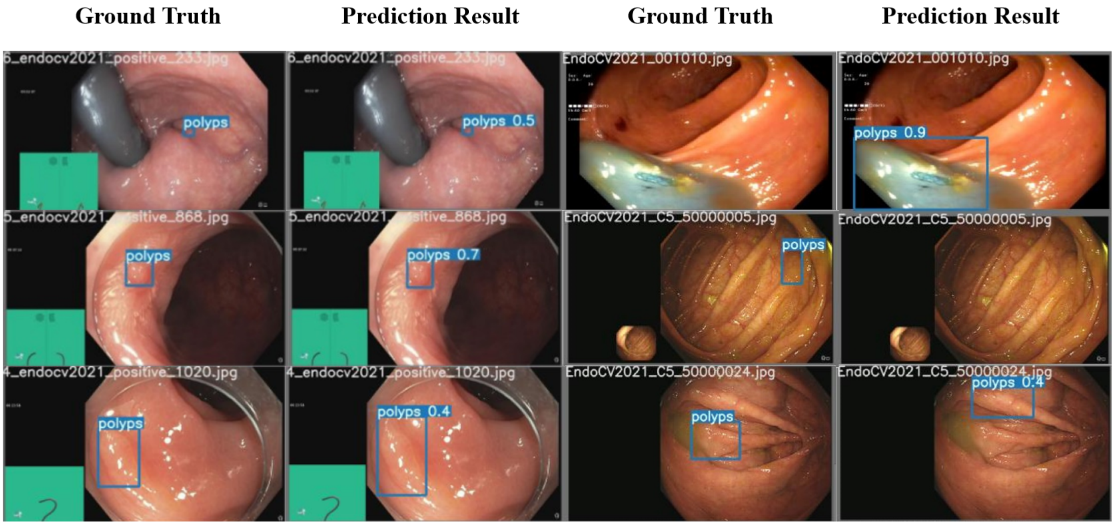
Ground Truth
Prediction Result
Ground Truth
Prediction Result
5_endocv2021_positive_233.jpg
polyps
5_endocv2021_positive_233.jpg
polyps 0.5
EndoCV2021_001010.jpg
EndoCV2021_001010.jpg
polyps 0.9
5_endocv2021_positive_868.jpg
polyps
5_endocv2021_positive_868.jpg
polyps 0.7
EndoCV2021_C5_50000005.jpg
polyps
EndoCV2021_C5_50000005.jpg
4_endocv2021_positive_1020.jpg
polyps
4_endocv2021_positive_1020.jpg
polyps 0.4
EndoCV2021_C5_50000024.jpg
polyps
EndoCV2021_C5_50000024.jpg
polyps 0.4

Figure 4: Ground truths and prediction results on some images from the validation dataset. The left two columns show the good result for polyps with different sizes. The right two columns show the bad result of false positive, false negative and error location.

Table 1
Results of baselines on EndoCV2021 detection task validation set

Baseline	Img-size	mAP:0.5:0.95
YOLOv5x-P6	1280×1280	0.5635
ScaledYOLOv4-P6	1280×1280	0.5601
EfficientDet-D6	1280×1280	0.4453

5. DISCUSSION & CONCLUSION

EndoCV2021 is an international endoscopy computer vision challenge for polyp detection and segmentation. For the task of polyp detection, we evaluated three state-of-the-art detection architectures: EfficientDet, ScaledYOLOv4 and YOLOv5. As a result, YOLOv5 was chosen as the baseline of our method because of the relatively better performance. The size of the training dataset is small, which has only 2910 images. To avoid over-fitting and improve the generalization ability of the model, several data augmentation methods were used, such as flipping, scaling, shearing, mixup, and mosaic. In addition, post-processing methods such as TTA and model ensemble were also used to improve the performance of the model. Finally, our method achieved state-of-the-art results on the test set. The score reached 0.7948 on the round-I test set, and 0.8824 on the round-II test set.

According to the ablation experiment, the performance of the model on small and medium objects showed obvious improvement when mosaic and mixup were used. This is because the original training set has only a small number of small objects. Mosaic and mixup methods could increase the number of small objects, which make the dataset more balanced. Besides, multi-scale training could improve the robustness of the model to input of different sizes. And

Table 2
Ablation experimental results on EndoCV2021 detection task validation set

Method	mAP:0.5:0.95	APsmall	APmedium	APlarge
YOLOv5x-P6	0.5324	0.2134	0.4577	0.5832
YOLOv5x-P6 **Mosaic, Mixup**	0.5635	0.2721	0.5182	0.5923
YOLOv5x-P6 Mosaic, Mixup + **Multi-scale Training**	0.6608	0.2877	0.4662	0.7075
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training **TTA**	0.6843	0.3453	0.4768	0.7334
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training TTA + **Ensemble (7 models)**	0.7107	0.2688	0.5387	0.7533
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training TTA + **Ensemble (20 models)**	0.7178	0.2361	0.5304	0.7629

Table 3
Results on EndoCV2021 detection task test set

Method	Test Phase	IoU-thres	Score	Speed
YOLOv5x-P6 Mosaic, Mixup + Multi-scale traing TTA + Ensemble (7 models)	Round I	0.6	0.7948±0.8375	0.725 s/img
YOLOv5x-P6 Mosaic, Mixup + Multi-scale traing	Round II	0.6	0.7696±0.2331	0.035 s/img
YOLOv5x-P6 Mosaic, Mixup + Multi-scale training TTA	Round II	0.6	0.8062±0.1250	0.104 s/img
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training TTA + Ensemble (7 models)	Round II	0.6	0.8322±0.2785	0.722 s/img
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training TTA + Ensemble (12 models)	Round II	0.6	0.8611±0.2292	1.232 s/img
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training TTA + Ensemble (20 models)	Round II	0.6	0.8615±0.2954	2.148 s/img
YOLOv5x-P6 Mosaic, Mixup + Multi-scale Training TTA + Ensemble (20 models)	Round II	0.3	0.8824±0.2743	2.056 s/img

because images with large objects occupy a large proportion in the dataset, APlarge on the validation set significantly increased when multi-scale training was used.

In addition, TTA and model ensemble were also helpful to improve the performance of the model, because they both combined multiple calculation results, which reduced the error of the model. However, we noticed that after using model ensemble, the performance on small objects became worse. This might be caused by the ensemble strategy that we chose was based on non-maximum suppression. For the detection of one small object, it's easier to get two bounding boxes from two models which had small IoU. So it would be detected as two different objects, which made the result worse. Furthermore, the results on the test set suggested that appropriately reducing the IoU threshold of the non-maximum suppression processing could also improve the performance of the model. It could be caused by the small IoU of the bounding boxes in the dataset.

Although our model has achieved state-of-the-art results on the test set, it has some unresolved problems. For example, it is time consuming for inference due to the usage of TTA and model ensemble. The model with the best performance on the round-II test set requires 2.056s/img for inference. In addition, it is difficult to use hyperparameter evolution algorithm based on this dataset due to the time limitation of the challenge. Moreover, according to [11], the masks provided by the dataset for segmentation can also be used to help train the detection model, which may also improve the performance of our method.

References

[1] P. Wang, X. Xiao, J. R. G. Brown, T. M. Berzin, M. Tu, F. Xiong, X. Hu, P. Liu, Y. Song, D. Zhang, et al., Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy, Nature biomedical engineering 2 (2018) 741–748.

[2] Y. Mori, S.-e. Kudo, M. Misawa, K. Mori, Simultaneous detection and characterization of diminutive polyps with the use of artificial intelligence during colonoscopy, VideoGIE 4 (2019) 7.

[3] M. Misawa, S.-e. Kudo, Y. Mori, T. Cho, S. Kataoka, A. Yamauchi, Y. Ogawa, Y. Maeda, K. Takeda, K. Ichimasa, et al., Artificial intelligence-assisted polyp detection for colonoscopy: initial experience, Gastroenterology 154 (2018) 2027–2029.

[4] S. Ali, D. Jha, N. Ghatwary, S. Realdon, R. Cannizzaro, M. A. Riegler, P. Halvorsen, C. Daul, J. Rittscher, O. E. Salem, D. Lamarque, T. de Lange, J. E. East, Polypgen: A multi-center polyp detection and segmentation dataset for generalisability assessment, arXiv (2021).

[5] S. Ali, M. Dmitrieva, N. Ghatwary, et al., Deep learning for detection and segmentation of artefact and disease instances in gastrointestinal endoscopy, Medical Image Analysis (2020). doi:https://doi.org/10.1016/j.media.2021.102002.

[6] S. Ali, F. Zhou, B. Braden, et al., An objective comparison of detection and segmentation algorithms for artefacts in clinical endoscopy, Scientific reports 10 (2020) 1–15.

[7] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.

[8] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Scaled-YOLOv4: Scaling cross stage partial network, arXiv preprint arXiv:2011.08036 (2020).

[9] G. Jocher, Yolov5, https://github.com/ultralytics/yolov5, 2021.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick,

Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[11] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, B. Zoph, Simple copy-paste is a strong data augmentation method for instance segmentation, arXiv preprint arXiv:2012.07177 (2020).