

CONIC: Method Description of Team Arontier

Heeyoung Ahn

Yiyu Hong

Department of R&D Center,
Arontier Co., Ltd, Seoul,
Republic of Korea

Department of R&D Center,
Arontier Co., Ltd, Seoul,
Republic of Korea

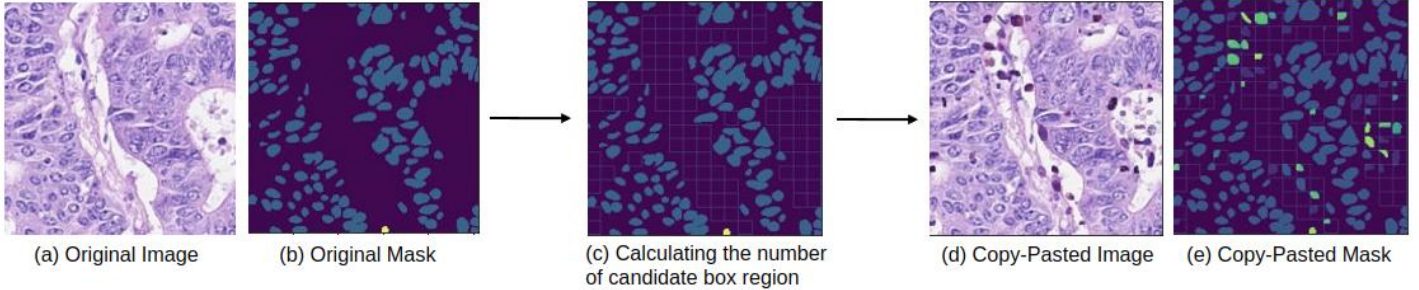


Figure 1. We calculate the number of candidate box region on which cells would be pasted and paste random cells according to the class ratio we set.

I. METHOD

This manuscript describes submitted method of team Arontier to CONIC Challenge [1-2].

Our method mainly focus on how to deal with the class imbalance using copy-paste augmentation [1] with a little bit modification. We increased the number of cell objects for insufficient classes (Neutrophil, Plasma, Eosinophil) intensively and for those sufficient classes weakly using copy-paste augmentation, respectively (Figure. 1). We pasted the cell objects to where they would not be overlapped with other originally existed cells.

A. Extracting all cell objects in advance

Unlike copy-paste augmentation[3] which do copy-paste in 2 different image pair, we want to paste cells from any images to target image. We extracted all cell objects from all training set and saved them in a file in advance. So that we can efficiently using these cell objects and pasting them on target train images when learning our model on the fly.

B. Source of pasted cells

For our copy-paste augmentation, we consider two method. One is source-level and the other is random-level. In source-level, we consider that there are five image sources(i.e., consep, crag, dpath, glas, pannuke) in datasets and we paste cells from image of specific source to other target image of same source. For example, we extract cells from images of consep source and pasted those cells to target images of same consep source.

Unlike this, in random-level we extract cells from images of all sources and paste those cells to target images randomly without considering any sources. In our experiments, we found that random-level yields better performance over source-level.

C. Calculate candidate box region

For pasting cells to non-overlapping region, we firstly calculate the empty candidate box region to which the cells would be pasted. To this end, we stride a window in target image and calculate the total number of candidate box region to which cells would be pasted. Considering various size of cells, we select window size with 15-pixel width and height.

D. Number of candidate box region to use and assigning different ratio for each class

After calculating the total number of candidate box region in each image, we set the ratio of how many boxes to use. We select random values of 0.1 interval between 0.1 and 1. Then we assign different ratio for each class according to the number of cells of a particular class in training images. Specifically, we assign different ratio of 0.3, 0.1, 0.05, 0.15, 0.3, 0.1 for each class because we want to deal with the insufficiency of particular classes. Finally if the pasted cells still overlap with other cells, we skip those cells.

E. Augmentation for pasted cell

We experimented several augmentation method such as flip, scaling for cells which will be pasted. We found that scaling was not effective so we use horizontal, vertical flip.

II. TRAINING DETAILS

We follow the HoverNet [4] but with some differences. We replace the HoverNet model structure with Unet++ [5] and modify the structure so that model can output 3 outputs like HoverNet. We select EfficientNet-b7 [6] pretrained on ImageNet [7] for encoder of Unet++. For Image Augmentation, we choose ColorJitter, Flip, Rotate, Blur, Gaussian Noise, Cutout [8], Cutmix [9]. We select Radam [10] optimizer with learning rate of $1e-3$. During training, we downscale learning rate from $1e-3$ to $1e-5$ using step scheduler with 0.1 ratio every 100 epoch. For loss function, we choose same loss function that baseline of challenge [1] applied except for classification output. For classification output, we choose focal loss [11] with $\gamma = 2$, $\alpha = 1$. We train the model with batch size 8 for 300 epochs.

Finally, we ensemble outputs of 5 models which were trained in 5-Fold cross validation setting.

REFERENCE

- [1] Graham, Simon, et al. "CoNIC: Colon Nuclei Identification and Counting Challenge 2022." *arXiv preprint arXiv:2111.14485* (2021).
- [2] Graham, Simon, et al. "Lizard: A Large-Scale Dataset for Colonic Nuclear Instance Segmentation and Classification." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
- [3] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, Barret Zoph. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. In CVPR, 2021.
- [4] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, Nasir Rajpoot. HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images. In CVPR, 2018.
- [5] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In CVPR, 2018.
- [6] Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In ICML, 2019.
- [7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In IJCV, 2015.
- [8] Terrance DeVries, Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. In CVPR, 2017.
- [9] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In CVPR, 2019.
- [10] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, Jiawei Han. On the Variance of the Adaptive Learning Rate and Beyond. In ICLR, 2019.
- [11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár. Focal Loss for Dense Object Detection. In CVPR, 2017