FIBA: Frequency-Injection based Backdoor Attack in Medical Image Analysis Supplementary Material

Yu Feng[1,*] Benteng Ma[1,*] Jing Zhang[2] Shanshan Zhao[3] Yong Xia[1,†] Dacheng Tao[3,2]
[1] Northwestern Polytechnical University
[2] The University of Sydney [3] JD Explore Academy
{fengy,mabenteng} @mail.nwpu.edu.cn,jing.zhang1@sydney.edu.au
sshan.zhao00@gmail.com, yxia@nwpu.edu.cn,dacheng.tao@gmail.com

1. Training Details

Experiments on ISIC-2019 [4]. We train the model with Adam optimizer [8] on ISIC-2019 for 200 epochs. ResNet50 [6] is chosen as the backbone network. The input size is set as 224×224, and batch size is 64. The learning rate is set to 0.01 and divided by 10 every 50 epochs. We follow the standard image augmentation strategies including random horizontal flips, vertical flips, and rotations.

Experimnets on KiTS-19 [7]. For the segmentation task, a coarse-to-fine segmentation framework is used in our experiments. In the first stage, we train the ResUnet [5] with Adam optimizer to coarsely segment the ROI regions which contains the whole kidney areas with cross-entropy loss at the first stage for 50 epochs. In the second stage, we train the DenseUnet [9] with Adam optimizer to segment the target areas of the tumor and kidney from the ROI regions with Dice loss [11]. The number of training epochs is 50, and the batch size is 6. During the training in both stages, the learning rate is set to 1e-4 and divided by 10 if the loss does not decrease. We also employ the horizontal flip augmentation strategy in both stages.

Experiments on EAD-2019 [1]. For the detection task on EAD-2019, we take Faster R-CNN [13] in mmdetection framework [2] with ResNet50 [6] as the backbone network, and we follow the default setting for training and evaluation. Specifically, we train the detection model with the SGD optimizer for 30 epochs. The learning rate is set to 0.005 and the batch size is 4. The input size is set as 512×512. We also employ random flip for data augmentation.

2. Hyper-parameter Study

There are two hyper-parameters in our method FIBA. One is the blended ratio α and the other one is β which determines the location and range of the low-frequency patch inside the amplitude spectrum to be blended. We investigated the influence of the two hyper-parameters on ISIC-2019 and KiTS-19 datasets.

Table 1. Results with different settings of α on ISIC-2019.

α	BA (%)↑	ASR (%)↑
0.05	85.15 ± 0.40	94.90 ± 0.61
0.10	85.15 ± 0.52	98.46 ± 0.29
0.15	85.43 ± 0.40	99.53 ± 0.08
0.20	85.50 ± 0.42	99.49 ± 0.10

Table 2. Results with different settings of β on ISIC-2019.

β	BA (%)↑	ASR (%)↑
0.05	85.17 ± 0.12	99.09 ± 0.17
0.10	85.43 ± 0.40	99.53 ± 0.08
0.15	84.90 ± 0.05	99.37 ± 0.16
0.20	85.24 ± 0.67	99.27 ± 0.20

We first conduct experiments with different blend ratio α on ISIC-2019. In Tab. 1, BA slightly increases with the growth of α while ASR peaks at a blend ratio 0.15. The poisoned images with different α are shown in Fig 2. We then investigate the impact of β in $\mathcal{M}$ with different values (*i.e.*, 0.05, 0.10, 0.15, 0.20) on ISIC-2019. As shown in Tab. 2, the proposed FIBA achieves consistent and high ASR $> 99.00\%$ with different β.

We further analyze the impact of α and β on the segmentation task (KiTS-19). α is set to 0.1, 0.2, 0.3, and 0.4, and β is set to 0.05, 0.10, 0.15, and 0.20. From Tab. 6, we find that ASR continues to improve with the increase of α. The poisoned samples with different α are shown in Fig. 3. We can see that some abnormal shades will occur in the CT images when $\alpha > 0.2$. Therefore, we choose $\alpha = 0.2$ for

*Equal contribution. This work was done during an internship at JD Explore Academy.
†Corresponding author

experiments on KiTS-19. As shown in Tab. 7, ASR peaks at $\beta = 0.1$ (71.44%) and we set $\beta = 0.1$ by default in those experiments on KiTS-19.

3. Results with Different Trigger Images

We then investigate the influence of using different trigger images in FIBA. As shown in Fig. 1, we select the other three typical images, including gray (the first row), animal (the second row), and human (the third row), from COCO validation set as the trigger images. The results of using these three trigger images are presented in Tab. 5. As can be seen, the proposed FIBA achieves consistent and high ASR $> 99\%$ when using different trigger images. It shows the effectiveness of FIBA that it does not depend on a specific choice of the trigger image.

4. Results with other attacks on ISIC-2019

We further supplement some contrast experiments with other attack methods. **ISSBA** [10]: the triggers which are generated from a trigger generator are sample-specific. **FIBA-C**:In stead of the square mask used in Eq. (6), we take the outer circle of square mask as the circle mask to implement FIBA method. **FIBA-H**: A variant of the FIBA attack with the high- frequency trigger pattern. As shown in Table 3 , FIBA outperforms FIBA-H and ISSBA in terms of both BA and ASR, while FIBA and FIBA-C achieve comparable and high results.

Table 3. Results with different attacks on ISIC-2019.

Method	BA (%)↑	ASR (%)↑
ISSBA	84.43 ± 0.16	99.33 ± 0.06
FIBA-C	85.14 ± 0.49	99.31 ± 0.15
FIBA-H	84.38 ± 0.08	98.43 ± 0.05
FIBA	85.43 ± 0.40	99.53 ± 0.08

5. Resistance to DF-TND [14]

we evaluated DF-TND [14] against our FIBA and other attack methods. The results of logit increases (LI) for the target class are shown in Table 4. The smaller the value of LI, the harder for DF-TND to defend. It shows that our FIBA achieves the lowest LI of 6.72, beating other attacks.

Table 4. Results of DF-TND against different attacks.

Method	BadNet	Blended	WaNet	ISSBA	FIBA-H	FIBA
LI↓	60.44	130.43	10.54	43.79	10.66	6.72

6. Running Time

We compare the running time of Blended [3] and the proposed FIBA on ISIC-2019 and all the experiments are conducted on a GeForce RTX 2080TI GPU. In addition, both the FIBA and Blended are implemented with the same training details (*e.g.*, epochs, batch size, learning rate, *et al*) as described in Sec. 1). For the proposed FIBA method, the FFT and iFFT operations in the trigger injection function are time-consuming when we implement them on the CPU, *i.e.*, it takes 23 hours for training on ISIC-2019, while Blended only takes 12 hours for training on ISIC-2019. However, when we accelerate the FFT and iFFT calculations on the GPU (through cupy [12] library), the training time can be greatly reduced to 9.5 hours, which is even faster than Blended.

(a) (b) (c) (d) (e)

Figure 1. Results of using different trigger images in the proposed FIBA method. (a) An original image from ISIC-2019. (b) Different trigger images. (c) The amplitude spectrums of the corresponding trigger images. (d) The images poisoned by different trigger image. (e) The residual maps.

Table 5. Results of using different trigger images in the proposed FIBA method on ISIC-2019.

Trigger image	BA (%)↑	ASR (%)↑
Gray	85.41 ± 0.47	99.16 ± 0.13
Animal	85.34 ± 0.40	99.66 ± 0.06
Human	85.69 ± 0.73	99.38 ± 0.02

References

[1] 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. 1

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

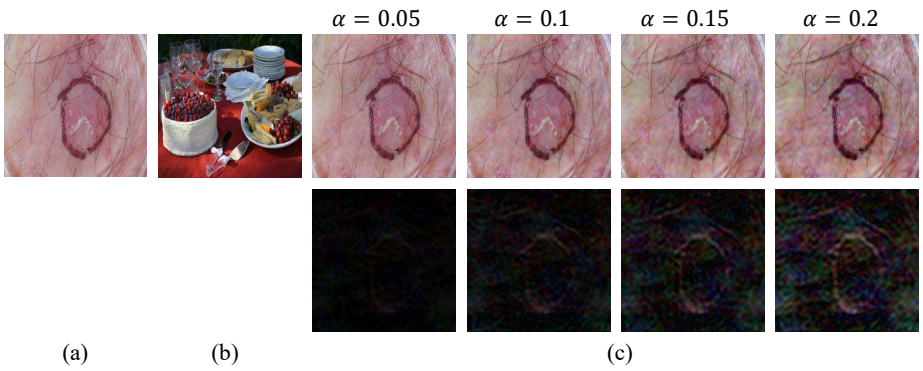

Figure 2. Visual comparison between different blended ratio α on ISIC-2019. (a) The original image. (b) The trigger image. (c) The poisoned images with different blended ratio α (upper row) and the residual maps (lower row).

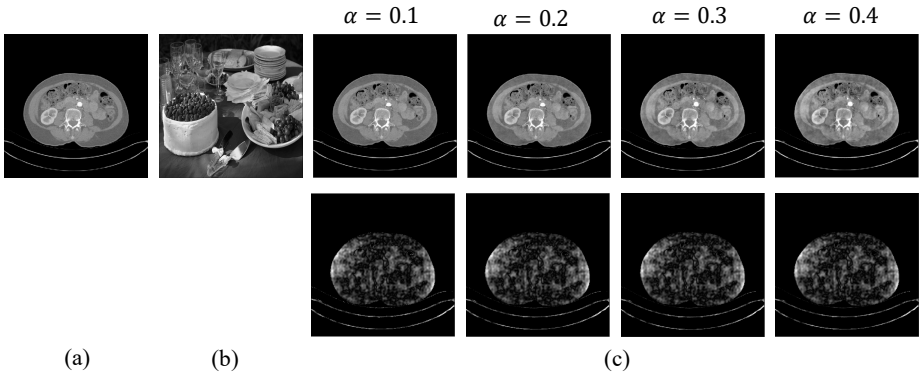

Figure 3. Visual comparison between different blended ratio α on KiTS-19. (a) The original image. (b) The trigger image. (c) The poisoned images with different blended ratio α (upper row) and the residual maps (lower row).

Table 6. Results with different settings of α on KiTS-19.

α	Clean data		Poisoned data		ASR (%)↑
	Organ(IoU)↑	Tumor(IoU)↑	Organ(IoU)↑	Tumor(IoU)↓	
0.1	93.75 ± 0.91	55.61 ± 4.27	93.46 ± 0.75	31.23 ± 4.21	58.83 ± 3.15
0.2	93.41 ± 1.12	54.54 ± 2.34	92.69 ± 1.17	21.02 ± 1.95	71.44 ± 4.90
0.3	93.11 ± 0.77	53.56 ± 3.32	92.35 ± 0.78	15.32 ± 5.77	75.41 ± 5.68
0.4	93.06 ± 0.61	52.50 ± 5.05	91.81 ± 0.83	11.59 ± 3.49	78.21 ± 3.51

Table 7. Results with different settings of β on KiTS-19.

β	Clean data		Poisoned data		ASR (%)↑
	Organ(IoU)↑	Tumor(IoU)↑	Organ(IoU)↑	Tumor(IoU)↓	
0.05	93.51 ± 0.85	55.12 ± 1.5	93.11 ± 0.81	21.93 ± 8.11	68.63 ± 8.21
0.10	93.41 ± 1.12	54.54 ± 2.34	92.69 ± 1.17	21.02 ± 1.95	71.44 ± 4.90
0.15	93.61 ± 0.87	54.79 ± 3.05	92.89 ± 0.79	20.83 ± 4.62	69.11 ± 5.72
0.20	93.51 ± 0.97	55.63 ± 2.4	92.35 ± 0.42	20.23 ± 5.32	69.31 ± 4.88

[3] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2

[4] Marc Combalia, Noel CF Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C Halpern, Susana Puig, et al. Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint*

arXiv:1908.02288, 2019. 1

[5] Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, 2020. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[7] Nicholas Heller, Niranjan Sathianathen, Arveen Kalapara, Edward Walczak, Keenan Moore, Heather Kaluzniak, Joel Rosenberg, Paul Blake, Zachary Rengel, Makinna Oestreich, et al. The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes. *arXiv preprint arXiv:1904.00445*, 2019. 1

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[9] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from CT volumes. *IEEE Trans. Medical Imaging*, 37(12):2663–2674, 2018. 1

[10] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021. 2

[11] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 1

[12] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *LearningSys@NeurIPS*, 2017. 2

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015. 1

[14] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *ECCV 2020*, 2020. 2