Supplementary Material: Multi-domain Learning for Updating Face Anti-spoofing Models

Xiao Guo, Yaojie Liu, Anil Jain, and Xiaoming Liu

Michigan State University
{guoxia11,liuyaoj1,liuxm,jain}@cse.msu.edu

In this supplementary material, we provide some detailed explanations on our method and implementation details in the experiment section.

1 More Method Details

1.1 Face Reconstruction Analysis

The motivation of using face reconstruction in our method is that, although it cannot reconstruct the perfect live faces given its spoof counterpart, but it can largely shed the light on spatial pixel location where spoofness occurs. As introduced in Sec. 3.2, we use such a reconstruction method to generate the preliminary mask $\mathbf{I}_{pre}$ as the pseudo label that supervises the proposed *SRE*. We offer the detailed visualization in Fig. 1.

In general, we have categorized spoof types into three main genera: (a) covered materials; (b) makeup stroke; (c) visual artifacts (*i.e.*, color distortion and moire effect) in the *replay* and *print* attacks. In particular, for covered materials, reconstruction methods largely erase these spoof materials, such as *funny glasses*, and *human mask*. As shown from Fig. 1, $\mathbf{I}_{pre}$ can roughly locate the pixel-wise spatial location that has been covered by the spoof material, and the estimated spoof region gives the more accurate prediction on pixels that are covered by these spoof materials. For makeup stroke, the reconstruction method changes the color and texture of the facial makeup area, making them similar to the natural skin. $\mathbf{I}_{pre}$ offers the scattered, discrete binary mask and estimated spoof region provides the smoother region indicating the spoofness. For *replay* and *print* attacks, the reconstruction method modifies facial structure (*i.e.,* nose and eyes) of the human face, or largely change the image's appearance, by providing the image with a sense of depth. Similar as makeup stroke genera, $\mathbf{I}_{pre}$ gives very discontinuous predictions on spoofness whereas the estimated spoof region is smoother and semantic.

1.2 Model Response

When a target domain image $\mathbf{I}_{target}$ is fed to the pre-trained model, the pre-trained model will be activated, as if the pre-trained model takes as input source domain images $\mathbf{I}_{source}$ which has resemblance with $\mathbf{I}_{target}$. In other words, the pre-trained model recognizes it as source domain images $\mathbf{I}_{source}$ which has common characteristic and pattern with $\mathbf{I}_{target}$. Therefore, source data can manifest

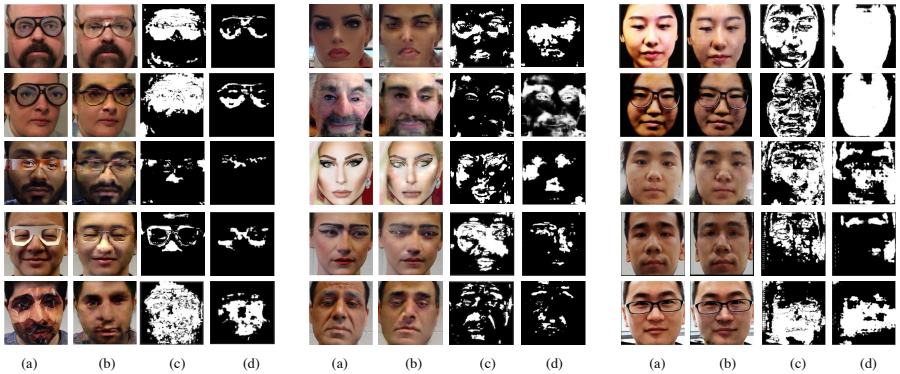

Fig. 1. The visualization of (a) input spoof image, (b) live counterpart reconstruction, (c) generated preliminary mask, and (d) estimated spoof region from our method. Three columns (from left to right) represent three different spoof genera, covered material, makeup stroke and visual artifacts from *replay* and *print* attack.

themselves on the response of the model, or in other words, keeping model response allows us to have memory or characteristics of the source data. With the development of deep learning, model can generate valid response in different applications [4, 5, 7, 1].

2 Experimental Results

2.1 PhySTD method details

In the experimental section, we apply FAS-*wrapper* on PhySTD for the analysis in Sec.5.2. In this section, we introduce the detailed architecture of PhySTD, which is illustrated in Fig. 2. Given the input image, PhySTD predicts spoofness, and disentangle the spoof trace into additive traces and inpainting components (*e.g.*, inpainting trace and region trace). Specifically, we firstly decompose the input image into three elements which represent the image information at different frequency levels. The feature extractor takes the concatenation of these three elements and generate multi-scale features which are fed to depth estimation branch for estimating the image depth, and three trace branches for estimating inpanting trace, region trace, and additive trace, respectively. Such three traces can be used to synthesize the live image, and this synthesized live image along with the genuine live image are fed to a multi-scale discriminator which is trained through the adversarial training.

2.2 The implementation details of prior methods

We are the first work that studies MD-FAS, in which no source data being available during the model updating process. To the best of our knowledge,

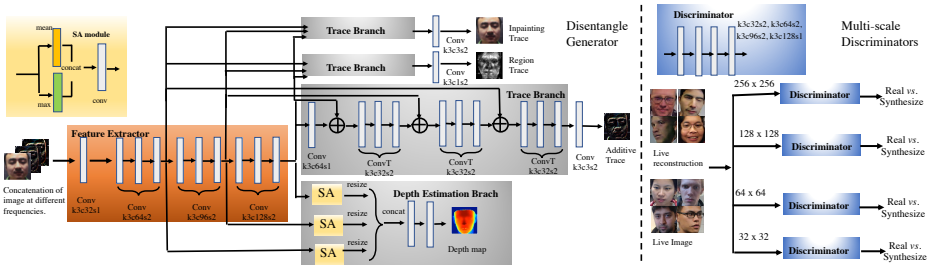

Fig. 2. The detailed architecture of PhySTD. The overall architecture contains Disentangle Generator and Multi-scale Discriminators. Notably, in the architecture of PhySTD, each convolutional layer is followed by Batch Normalization layer, RELU activation function and Dropout. This level of details is not included here.

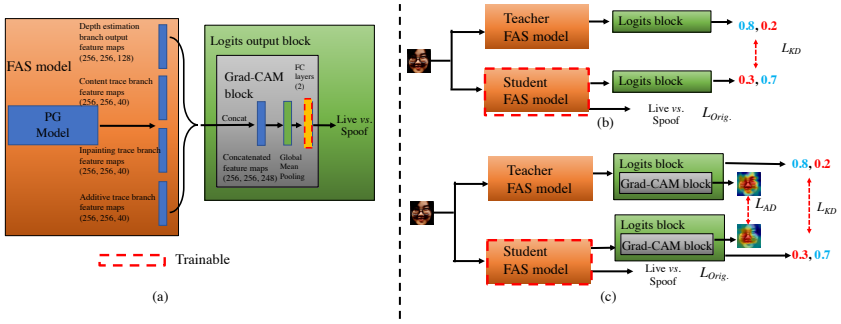

Fig. 3. In (a), we modify the pre-trained FAS model into a binary classifier. In (b) and (c), we modified architectures for LwF and LwM methods.

there does not exist FAS works in such a source-free scenario. Therefore, in order to have a fair comparison, we need to implement methods from other topics (*e.g.*, *anti-forgetting learning* and *multi-domain learning*) on FAS dataset. In this section, we explain our implementation details on prior methods.

The implementation details of prior anti-forgetting methods We compare our methods to prior works that have anti-forgetting mechanism: LwF [8], LwM [3] and MAS [2]. Firstly, we pre-train the FAS model that is based on PhySTD on the source domain dataset. After the pre-training, we concatenate output feature maps generated from the last convolution layer in different branches as a new concatenated feature maps. Then we feed such feature maps through Global Average Pooling Layer and a fully-connected (FC) layer, such that we can obtain a binary classifier. The details are depicted in Fig. 3(a). We fix the pre-trained FAS model weights and train the last FC layer. As a result, we can use concatenated feature maps for a binary classification result indicating spoofness. We denote newly-added layers as the Logits block, part of which generates the class activate map is denoted as Grad-CAM block. We use these two blocks with the original FAS model for implementing LwF and LwM, as

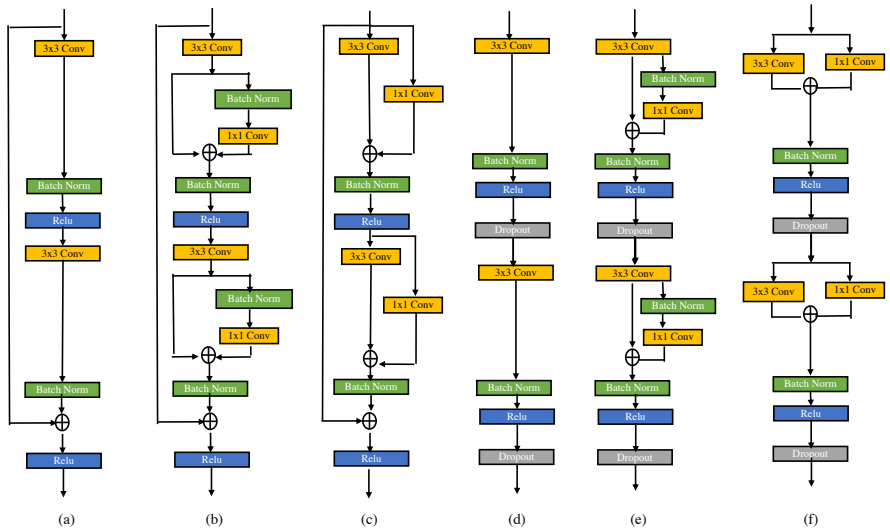

Fig. 4. Based on the RseNet building block (a), [9, 10] have proposed RseNet modified building blocks in (b) and (c) for learning multiple domain knowledge. Likewise, given two consecutive building blocks in PhySTD, we construct modified building blocks, based on [9, 10], in (e) and (f).

illustrated in Fig. 3 (b)(c). In terms of MAS [2], we apply the publicly available source code [1] on the binary classifier we construct, without significantly changing the architecture.

The implementation details of multi-domain learning methods Seri. Res-Adapter [9], and Para. Res-Adapter [10] are proposed for learning knowledge in multiple visual domains. Specifically, they use domain-specific adapter to enhance model ability in learning a universal image representation for multiple domains. They design such an idea on ResNet [6], which can be seen in Fig. 4. Based on the same idea, we modify the building block in PhySTD for learning the new domain knowledge. Notably, we have examine different adapter architectures, such as convolution filter with kernel size 1×1, 3×3, 5×5 and 7×7 , and find that 1×1 convolution offers the best FAS performance. We also consider the publicly available source code [2] as the reference for the implementation.

[1] https://github.com/rahafaljundi/MAS-Memory-Aware-Synapses
[2] https://github.com/srebuffi/residual_adapters

References

1. AbdAlmageed, W., Mirzaalian, H., Guo, X., Randolph, L.M., Tanawattanacharoen, V.K., Geffner, M.E., Ross, H.M., Kim, M.S.: Assessment of facial morphologic features in patients with congenital adrenal hyperplasia using deep learning. JAMA network open (2020)
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 139–154 (2018)
3. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5138–5146 (2019)
4. Guo, X., Choi, J.: Human motion prediction via learning local structure representations and temporal dependencies. In: AAAI (2019)
5. Guo, X., Mirzaalian, H., Sabir, E., Jaiswal, A., Abd-Almageed, W.: Cord19sts: Covid-19 semantic textual similarity dataset. arXiv preprint arXiv:2007.02461 (2020)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Hsu, I., Guo, X., Natarajan, P., Peng, N., et al.: Discourse-level relation extraction via graph pooling. arXiv preprint arXiv:2101.00124 (2021)
8. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
9. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. arXiv preprint arXiv:1705.08045 (2017)
10. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8119–8127 (2018)