2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

CORRECTING PSEUDO LABELS WITH LABEL DISTRIBUTION FOR UNSUPERVISED DOMAIN ADAPTIVE VULNERABLE PLAQUE DETECTION

Peiwen Shi *Jingmin Xin* *Nanning Zheng*

National Engineering Laboratory of Visual Information Processing and Applications
and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

ABSTRACT

Pseudo-label-based unsupervised domain adaptation (UDA) has increasingly gained interest in medical image analysis, aiming to solve the problem of performance degradation of deep neural networks when dealing with unseen data. Although it has achieved great success, it still faced two significant challenges: improving pseudo labels' precision and mitigating the effects caused by noisy pseudo labels. To solve these problems, we propose a novel UDA framework based on label distribution learning, where the problem is formulated as noise label correcting and can be solved by converting a fixed categorical value (pseudo labels on target data) to a distribution and iteratively update both network parameters and label distribution to correct noisy pseudo labels, and then these labels are used to re-train the model. We have extensively evaluated our framework with vulnerable plaques detection between two IVOCT datasets. Experimental results show that our UDA framework is effective in improving the detection performance of unlabeled target images.

Index Terms— Unsupervised domain adaption, pseudo label, label distribution, plaque detection, IVOCT.

1. INTRODUCTION

Vulnerable plaques are the leading cause of acute coronary syndrome, which can be successfully detected on IVOCT images by the deep learning-based method [1]. However, the trained deep model often meets performance degradation due to the following reasons: 1) detecting images obtained from different imaging equipment manufacturers; 2) detecting images obtained from different hospitals. However, these images have the same visual appearance and are not problematic for doctors to diagnose. This situation is known as the domain shift, which is one of the critical factors that prevent the transfer of research results into real-world applications. One straightforward idea is to annotate more training data of the target environments and then re-train the neural network. However, annotating data for each new domain is time-consuming and expensive or sometimes even infeasible, especially in the medical field that requires expert knowledge. To address the challenge, the researchers resort to unsupervised domain adaption (UDA) [2], for which no labels in the target domain are required.

The main challenge of UDA is the discrepancy of data distribution between the source domain and target domain. Current methods involve two lines of methods, domain-translation-based UDA and pseudo-label-based UDA. Domain translation-based UDA methods adopt [3] the image-to-image translation model to translate source-domain images to have the same style as the target-domain images while retaining their original contents. Then the model is adapted to the target domain by training with such domain-translated images and their ground-truth labels. However, this line of work is sub-optimal. Because the image-to-image translation model, Generative Adversarial Networks (GAN)[4], is only independently trained and does not measure the similarities between target-domain images correctly. Pseudo-label-based UDA [5] methods focus on learning target-domain features with generated pseudo labels, which are implemented by a two-stage training scheme: (1) supervised pre-training on the source domain with ground-truth labels, and (2) unsupervised fine-tuning on the target domain with pseudo labels. However, pseudo labels usually suffer from the noise caused by the model trained on the different data distribution. The noisy label could damage the subsequent learning. There are two methods to solve this problem: improving the precision of pseudo labels and mitigating the effects caused by noisy pseudo labels. Some existing works have proposed to set the threshold to neglect the low-confidence pseudo labels to improve the precision of pseudo labels [6]. Although pseudo labels' precision is improved to a certain degree, it still exists noise in the pseudo labels.

We propose an end-to-end method for UDA via label distribution learning to address the mentioned challenges, which could transfer a fixed categorical value to distribution and correct noisy pseudo labels by updating the label distributions to obtain a robust training model. Without introducing extra modules, we transfer the fixed noisy labels to a distribution. Then, we iteratively update the network parameters and the label distribution. Therefore, the proposed method could effectively exploit the target-domain information offered by corrected pseudo labels and take advantage of the unlabeled

This work was supported by National Key R&D Program of China Grant 2017YFA0700800.

target-domain data, which is critical in deploying the trained model in clinical practice. We perform experiments on the 2017 IVOCT dataset and Harbin-oct dataset to investigate the UDA of vulnerable plaque classification. It obtains comparable performance with the supervised learning methods on the target domain.

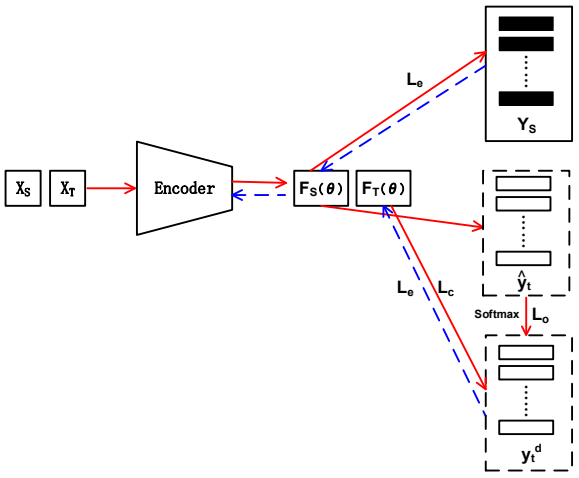

Fig. 1. The UDA framework. We first train the source-domain model with supervised ground-truth label(Y_S). Then, noisy pseudo labels ($\hat{y}_t$) of target domain dataset can be obtained by prediction results of source-domain model. Next, we use label distribution (y_t^d) to replace noisy pseudo labels ($\hat{y}_t$) by softmax function, and the label distribution are updated in every iteration using three loss function. Last, we use correct pseudo labels to fine-tune the model. The red arrow indicates forward propagation, and the blue arrow indicates backward propagation.

2. METHOD

In this section, we first provide the definition and representation of the problem. Then, we revisit the conventional domain adaptive method based on the pseudo label, and the label distribution learning is proposed to mitigate the effects caused by noisy pseudo labels. Finally, a training strategy is given to optimize the whole framework.

2.1. Problem Definition

Given the labeled dataset $X_s = \{x_{si}\}_{i=1}^{M}$ from the source domain and the unlabeled dataset $X_t = \{x_{tj}\}_{j=1}^{N}$ from the target domain where M and N denote the number of labeled source data and the unlabeled target data. The unsupervised classification domain adaption intends to learn a model F from the source domain, then estimate the model parameter θ_t to minimize the prediction bias on the target-domain inputs. During training, the label $Y_s = \{y_{si}\}_{i=1}^{M}$ of source domain dataset is provided while the target-domain label $Y_t = \{y_{tj}\}_{j=1}^{N}$ remains unknown. We minimize the cross-entropy loss in Equation 1, then the discrepancy between predicted results and the ground-truth probability on the source-domain dataset is minimized.

$$Bias(y_t) = E[-y_{tj} log F(x_{tj}|\theta_t)] \tag{1}$$

where y_{tj} is the ground-truth label, and $F(x_{tj}|\theta_t)$ is the predicted probability of x_{tj}. During testing, we utilize the trained model F to predict the label of test examples.

2.2. Pseudo Label Learning

Pseudo label learning is to leverage the pseudo label to learn feature representation from the unlabeled data, which tackled unsupervised domain adaption following a two-stage training scheme: (1) supervised pre-training on the source domain, and (2) unsupervised fine-tuning on the target domain. Pseudo labels could be obtained via the model of stage one: $\hat{y}_t^j = argmax F(x_t^j|\theta_s)$ where θ_s is the model parameters learned from the source-domain training data. Then the prediction bias is minimized on stage two with Equation 1 to obtain the target model. Existing methods consider pseudo labels $\hat{y}_t$ as true labels and train the model parameters θ_t to minimize the bias between the prediction and pseudo labels. However, because of different data distribution between X_s and X_t, the obtained pseudo labels $\hat{y}_t$ are not accurate. Moreover, noisy pseudo labels can lead to serious overfitting and dramatically reduce the trained network's accuracy.

2.3. Label distribution learning

Inspired [7], we utilize label distribution learning to address the label noise. We translate noisy pseudo labels to label distribution, and this probabilistic setting allows ample flexibility for noise correction. The pseudo label $\hat{y}_t$ can be translated to y_t^d and $y_{tj}^d \in S = \{y : y \in [0,1]^c, 1^T y = 1\}$ for every image x_{tj}, and y_{tj}^d is our estimate of the underlying noise-free label for x_t^j, which is initialized based on the noisy pseudo label $\hat{y}_{tj}$. Then, it is continuously updated through back-propagation.

The label distribution y_{tj}^d models the unknown noise-free label for x_{tj}. Hence, we need to estimate these distributions in our learning process. We let y_{tj}^d be part of the updated parameters during the back-propagation process. We not only update the network parameters θ_t but also update y_t^d in every iteration. Therefore, we optimize both network parameters and label distribution as follows:

$$\min_{\theta_t, y_t^d} L(\theta_t, y_t^d | X_t) \tag{2}$$

The overall architecture is shown in Fig.1. After training, y_t^d will be a good estimate of the underlying unknown noise-free label (corrected label).

As mentioned above, we use noisy pseudo labels $\hat{y}_t$ to initialize label distribution y_t^d. However, the original noisy

label $\hat{y}_t$ does not directly affect the model ($F(x_{tj}|\theta_t)$) learning, and we use it to indirectly initialize our label distribution y_t^d, which can be denoted as follows:

$$y_t^d = softmax(K\hat{y}_t) \tag{3}$$

where K is a large constant (K=2 in our experiments).

We minimize the bias between our label distribution y_t^d and the network prediction $f(x_t;\theta)$ to guides how the network parameters should be updated. As same as the previous label distribution learning studies [7], we also use KL-loss to calculate the distance between these two distributions. Hence, Equation 2 can be reformulated as:

$$L_c(f(x_t;\theta_t), y_t^d) = \frac{1}{N}\sum_{j=1}^{N} KL(f(x_{tj};\theta_t)||y_{tj}^d) \tag{4}$$

We should notice that although the pseudo labels contain noise, they also have lots of correct labels. Therefore, we should not let the estimate label distribution y_t^d totally different from those noisy labels $\hat{y}_t$. Then, we introduce a compatibility loss $L_o(\hat{y}_t, y_t^d)$ to ensure this requirement, which can be denoted as:

$$L_o(\hat{y}_t, y_t^d) = -\frac{1}{N}\sum_{j=1}^{N} \hat{y}_{tj} log y_{tj}^d \tag{5}$$

which is a classic cross-entropy loss between label distribution and noisy label.

Because of label distribution y_t^d as supervision signal, the model $f(x_t;\theta_t)$ tend to approach y_t^d fairly quickly. We add a loss term following the previous work, named the entropy loss, to avoid this problem. The entropy loss can force the network to peak at only one category rather than flat because the one-hot distribution has the smallest possible entropy value. This property is advantageous for classification problems. The entropy loss is defined as

$$L_e(f(x_t;\theta_t)) = -\frac{1}{N}\sum_{j=1}^{N} f(x_{tj};\theta_t) log f(x_{tj};\theta_t) \tag{6}$$

Hence, the overall loss function is

$$L = L_c(f(x_t;\theta_t), y_t^d) + \alpha L_o(\hat{y_t^d}, y_t^d) + \beta L_e(f(x_t;\theta_t)), \tag{7}$$

in which α and β are two hyperparameters.

2.4. Training and Testing

During training, we first train the model using the provided ground-truth label and the cross-entropy loss on the source-domain dataset. Secondly, we use the trained source-model to predict the target-domain dataset and get pseudo labels. Thirdly, we utilize pseudo labels to initialize label distribution y_t^d, and then the y_t^d is used as true labels to update both network parameters and label distributions. According to the previous study, updating $\hat{y_t^d}$ requires a much larger learning rate than other parameters. Hence, we use a single hyperparameter λ to update y_t^d, as

$$y_t^d = softmax(K\hat{y}_t) \leftarrow K\hat{y}_t - \lambda\frac{\partial L}{\partial K\hat{y}_t} \tag{8}$$

At the end of this step, the corrected label distribution for each image is obtained. Finally, we use only the KL-loss(i.e., $\alpha = \beta = 0$) to fine-tune the network using the learned label distributions.

During testing, we use the fully trained network to preform prediction for future test examples.

3. EXPERIMENTS

We evaluate our proposed UDA method on two real-world datasets: IVOCT2017 and HarbinOCT. All experiments were implemented using the PyTorch framework.

3.1. Datasets and Metrics

IVOCT-2017 [8] dataset was provided by the 2017 Chinese conference on computer vision (CCCV)-IVOCT based vulnerable plaque detection challenge. It contained 3000 images and was divided into two parts: training data (2000 images, including 1000 positives and 1000 negatives), testing data (300 images, including 198 positives and 102 negatives). We also build a new dataset (HarbinOCT) based on clinical IVOCT images acquired with a St. Jude Ilumien OPTIS. The HarbinOCT consisted of 3106 images and was split off an independent test set of 309 images (153 positives and 156 negatives). In this paper, we train the model on one dataset and adapt it to another dataset. The ground-truth label of the target dataset is unknown.

The evaluation metrics of our UDA framework is based on three criteria, including precision (P), recall (R), and harmonic mean (F-score), which can be defined as:

$$\begin{cases} P = \frac{nTP}{nTP+nFP} \\ R = \frac{nTP}{nTP+nFN} \\ F-score = 2\frac{P\times R}{P+R} \end{cases} \tag{9}$$

where nTP, nFP, and nFN represent the number of true positives, false positives, and false negatives, respectively.

3.2. Implementation and Results

We use ResNet-18 as the backbone network for feature representation. And the initial learning rate is 0.01, $\alpha = 0.1, \beta = 0.4$, and $\lambda = 100$. Denoiseing, flattening and horizontal random flip were performed as data preprocessing and augmentation. We used SGD with 0.9 momentum, a weight decay of 10^{-4}, and a batch size of 20. We first trained the source model with a learning rate of 0.01 and divided it by 10 after every 8 epochs. Then, we fine-tuned the trained source model with

Table 1. Evaluate the performance of our UDA framework for VPS classification

IVOCT to HarbinOCT			
Method	P	R	F-score
Supervised training	0.8888	0.9934	0.9382
W/o adaption	0.6371	0.9869	0.7743
Pseudo-label [5]	0.8445	0.8169	0.8305
DANN [9]	0.8679	0.9019	0.8846
ADDA [10]	0.7219	0.9673	0.8268
Our UDA	0.875	0.9150	**0.8945**

Table 2. Evaluate the performance of our UDA framework for VPS classification

HarbinOCT to IVOCT			
Method	P	R	F-score
Supervised training	0.9361	0.8888	0.9119
W/o adaption	0.72	0.97	0.83
Pseudo-label [5]	0.7471	0.9494	0.8496
DANN [9]	0.7758	0.9090	0.8372
ADDA [10]	0.8729	0.7979	0.8337
Our UDA	0.8148	0.8888	**0.8502**

pseudo labels of the target dataset. The epochs numbers for the three stages were 15, 20, and 60. In the last step, we used the learning rate of 0.05 and divided it by 10 after 33 and 46 epochs.

To observe the effect of domain shift on classification performance, we first obtain the 'W/o adaption' lower bound by directly applying the model learned in the source domain to test target images without using any domain adaption method. We also provide the performance upper bound of supervised learning with target domain labels to measure the performance gap. For a consistent comparison, ResNet-18 is adopted for training lower and upper bounds. Furthermore, we compare our method with three UDA methods including Pseduo-label [5], DANN [9] and ADDA [10]. Tabel 1 and Tabel 2 reports that the model trained on the source dataset only obtained the F-score of 0.7743 and 0.83 when being tested on the target dataset directly. The significant performance gap to the supervised training upper bound is 28 and 8 percentage points, demonstrating the server domain gap between datasets from different sources and leading to a large degradation of performance. Remarkably, our UDA method improves the classification performance to 0.8945 and 0.8502 and overpasses three UDA methods. These quantitative results validate our method's effectiveness in addressing noisy pseudo labels and the severe domain shift.

4. CONCLUSION

In this paper, we have presented a novel pseudo-label-based UDA framework for VPS detection. We transferred the fixed pseudo labels in label distribution and iteratively update both network parameters and label distribution to gain robust target models. Experimental results demonstrate our UDA framework's effectiveness in improving VPS detection performance in the unlabeled target domain, which lays the foundation of the deployment of decision support systems in clinical practice. Future work focuses on solving the stability of the algorithm and simplifying the algorithm.

5. REFERENCES

[1] Ran Liu, Yanzhen Zhang, Yangting Zheng, Yaqiong Liu, Yang Zhao, and Lin Yi, "Automated detection of vulnerable plaque for intravascular optical coherence tomography images," *Cardiovascular engineering and technology*, vol. 10, no. 4, pp. 590–603, 2019.

[2] Mei Wang and Weihong Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[3] Cheng Chen, Qi Dou, Hao Chen, Jing Qin, and Pheng Ann Heng, "Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation," *IEEE Transactions on Medical Imaging*, 2020.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[5] Dong-Hyun Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," .

[6] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang, "Domain adaptation for semantic segmentation via class-balanced self-training," *arXiv preprint arXiv:1810.07911*, 2018.

[7] Kun Yi and Jianxin Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7017–7025.

[8] "Ivoct based the vulnerable plaque detection challenge in the 2017 chinese conference on computer vision (cccv-ivoct)," 2017.

[9] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.