Local and Global Consistency Regularized Mean Teacher for Semi-supervised Nuclei Classification

Hai Su[1], Xiaoshuang Shi[1], Jinzheng Cai[1], and Lin Yang[1,2](✉)

[1] Department of Biomedical Engineering,
University of Florida, Gainesville, FL 32611, USA
{hsu224,xsshi2015,jimmycai}@ufl.edu

[2] Department of Electrical and Computer Engineering,
University of Florida, Gainesville, FL 32611, USA
lin.yang@bme.ufl.edu

Abstract. Nucleus classification is a fundamental task in pathology diagnosis for cancers, *e.g.*, Ki-67 index estimation. Supervised deep learning methods have achieved promising classification accuracy. However, the success of these methods heavily relies on massive manually annotated data. Manual annotation for nucleus classification are usually time consuming and laborious. In this paper, we propose a novel semi-supervised deep learning method that can learn from small portion of labeled data and large-scale unlabeled data for nucleus classification. Our method is inspired by the recent state-of-the-art self-ensembling (SE) methods. These methods learn from unlabeled data by enforcing consistency of predictions under different perturbations while ignoring local and global consistency hidden in data structure. In our work, a label propagation (LP) step is integrated into the SE method, and a graph is constructed using the LP predictions that encode the local and global data structure. Finally, a Siamese loss is used to learn the local and global consistency from the graph. Our implementation is based on the state-of-the-art SE method *Mean Teacher*. Extensive experiments on two nucleus datasets demonstrate that our method outperforms the state-of-the-art SE methods, and achieves F_1 scores close to the supervised methods using only 5%–25% labeled data.

Keywords: Nucleus classification · Semi-supervised learning · Deep learning

1 Introduction

Nucleus type information is essential in many pathology diagnoses [4,9]. In many settings, the presence and portion of certain types of nucleus are used to assess the proliferation rate, subtypes or grade of the diseases [4,13]. Traditionally, nucleus classification is treated as a supervised classification problem [3,4,9]

D. Shen et al. (Eds.): MICCAI 2019, LNCS 11764, pp. 559–567, 2019.
https://doi.org/10.1007/978-3-030-32239-7_62

and deep neural networks have achieved rather satisfactory performance. However, the superiority of supervised deep learning usually heavily relies on the availability of massive manually annotated data. As well known that large- scale annotation for medical data is expensive and time consuming, *e.g.*, diagnostic pathology images, while large-scale unlabeled data are relatively easy to obtain. To alleviate the high demand for manual annotation, semi-supervised deep learning (SSDL) has been developed to learn from a small portion of labeled data and large-scale unlabeled data. Recently, self-ensembling (SE) based semi-supervised learning has attracted broad attention [7,8,11]. The intuition of SE method is to enforce a prediction consistency for each training sample under different perturbations. Such consistency is not dependent on label information, and is able to extract extra semantic information from the unlabeled data. One of the successful SE method is called *temporal ensembling* (TE) [5]. In TE, for each unlabeled sample, an exponential moving average (EMA) of the prediction within multiple previous training epochs is computed as the proxy target. A mean square error (MSE) between the predictions and the proxy targets is used as the consistency loss. The proxy targets are the ensembled predictions of those from many previous epochs, thus serve as stronger proxy labels that provide extra semantic information in addition to the labeled data. However, TE requires to maintain a matrix of size $N \times C$, where N denotes the number of training samples, including labeled and unlabeled data, and C is the number of classes. This requirement makes TE model heavy when learning on large datasets. To alleviate this problem, *Mean Teacher* (MT) [10] utilizes two models (student and teacher models). Instead of maintaining the EMA of the proxy labels, MT method maintains a teacher model as the EMA of the student model. In each minibatch evaluation, the output of the teacher model is used as the proxy target. Since such proxy target is generated by the EMA model aggregated from many student models, it provides better proxy targets.

One aspect ignored by the aformentioned SE methods is the intrinsic structure of data. That is the local and global consistency widely existing in many datasets [2,12]. Local consistency refers to that samples from the same class are likely to lie in the same vicinity in the feature space. Global consistency means that samples from the same global structure are likely to share the same label. To enforce the local and global consistency, in this paper, we propose a novel loss function that is computed over a graph constructed via label propagation (LP) [14]. Specifically, we utilize the LP algorithm to iteratively propagate the label information from the labeled samples to the unlabeled ones based on the local structure until a global stable state is reached, then construct a graph based on the LP predicted labels. Next, Siamese loss is employed to pull the data from same class closer and push those from different classes further away. Therefore, the two consistencies are enforced. Experiments on two nucleus classification datasets illustrate the superior performance of the proposed method over the recent state-of-the-art SE methods.

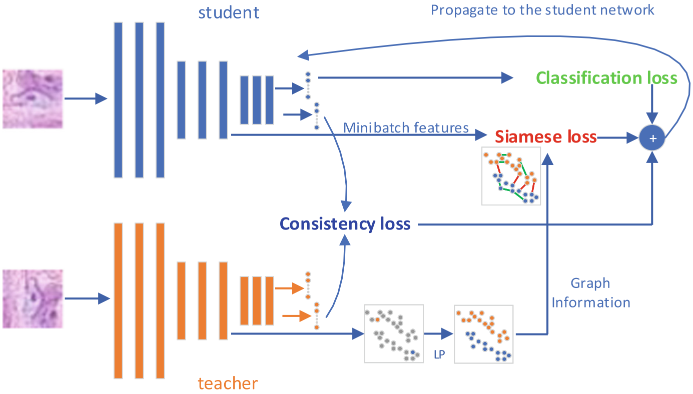


Fig. 1. Each minibatch consists of both labeled and unlabeled samples. The LP predicted labels and the ground truth labels are used to construct a graph capturing the local and global structure of the data. A Siamese loss is computed based on the graph. The student network is updated by a hybrid loss consisting of classification loss, consistency loss and the Siamese loss.

2 Mean Teacher with Label Propagation

2.1 Preliminaries

Since our method is based on mean teacher (MT) [10], we first briefly introduce mean teacher in this subsection. Let $\mathcal{X}_l = \{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}^m$ denote the labeled data and $\mathcal{X}_u = \{x_{n+1}, x_{n+2}, \cdots, x_N\} \subset \mathbb{R}^m$ denote the unlabeled data. The system consists of two networks, *i.e.*, the student network and the teacher network. The parameters of the teacher network is the EMA of the student network computed by: $\theta'_\tau = \alpha\theta'_{\tau-1} + (1-\alpha)\theta_\tau$, where α denotes the EMA coefficient, and θ and θ' represent the parameters of the student model and the teacher model, respectively. τ represents the global training iteration. The student network is updated by the following loss:

$$Loss_{mt} = \frac{1}{n}\sum_{i}^{n}(-y_i \log f_\theta(x_i)) + w(\tau)\lambda_{EMA}\mathbb{E}_{x,\eta,\eta'}[\|f_{\theta'}(x_j,\eta') - f_\theta(x_j,\eta)\|], \tag{1}$$

where λ_{EMA} is the coefficient controlling the strength of consistency between predictions of the same sample under different perturbations represented by η and η'. $w(\tau)$ is a ramp function of the global iterations τ. The first term is the cross-entropy loss for the labeled data and the second term enforces the consistency between the predictions of the student network $f_\theta(x,\eta)$ and the teacher network $f_{\theta'}(x,\eta')$. The consistency term is computed on all the data.

2.2 Local and Global Consistency Regularized Mean Teacher

As mentioned before, the MT method ignores the connection between the samples thus fails to extract more semantic information from the unlabeled data. In the proposed method, for each minibatch, LP is first conducted on the intermediate level features from the teacher network. This is because the teacher network is an ensemble model that is supposed to generate better feature embedding. Then a graph is constructed using the ground truth labels and the LP predicted labels. Next, a Siamese loss is calculated based on the graph using the features generated from the student network. Finally, a novel hybrid loss, including the loss Eq. (1) and the Siamese loss, is used to update the student network. An overview of our proposed system is depicted in Fig. 1.

Label Propagation: Label propagation [14] is a transductive semi-supervised learning algorithm. It propagates label information from the labeled data to the unlabeled data based on the affinity matrix of the data. The basic idea is that the data close to each other are more likely to share the same label. Therefore, the LP procedure computes the label of an unlabeled data as the weighted sum of the labels of its neighbors. Through an iterative procedure, the label can be propagated from the labeled data to their neighbors, and the neighbors of neighbors. Finally, the unlabeled data are assigned labels that respect the global structure of the data. The LP algorithm is proven to converge. More details of the proof can be found in [14].

Graph Based Clustering Loss: With the LP predicted labels for the unlabeled data, the pairwise connection information between the data points are known. With this information a graph can be built by:

$$A_{ij} = \begin{cases} 1, & \text{if } y_i = y_j, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

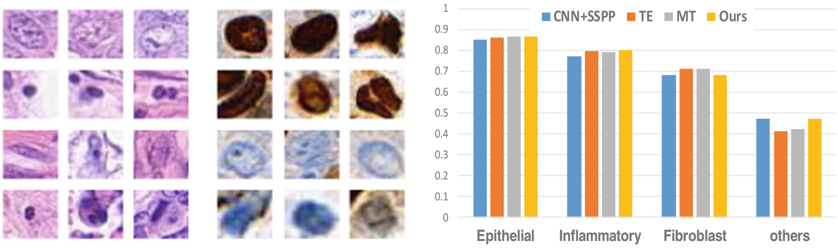
where y_i denotes the LP predicted labels for unlabeled data ($j \leq n < i \leq N$) and the ground truth labels for the labeled data ($i, j \leq n$). To enforce the local and global consistencies, we propose to use the contrastive Siamese loss [1] to pull the samples within the same class closer and push those from different classes further away:

$$L_s = \begin{cases} \|z_i - z_j\|^2, & \textit{if } A_{ij} = 1, \\ \max(0, m - \|z_i - z_j\|^2), & \textit{if } A_{ij} = 0, \end{cases} \tag{3}$$

where z_i represents the feature vector from the intermediate layers of the student network and m is a hyperparameter. The final proposed loss function is:

$$L_{total} = Loss_{mt} + w(\tau)(\lambda_{g1} \sum_{x_i, x_j \in \mathcal{X}_l} L_{s1} + \lambda_{g2} \sum_{x_i \in \mathcal{X}_l, x_j \in \mathcal{X}_u} L_{s2}), \tag{4}$$

where λ_{g1} denotes the weight of the Siamese loss computed on the labeled samples, and λ_{g2} represents the weight of the Siamese loss computed on both unlabeled and labeled data. Since the LP does not change the labels of the labeled samples, the Siamese loss L_{s1} ensures that there is always some correct information for learning. Note that we do not compute Siamese loss between the unlabeled samples. This is because the LP-predicted labels are very noisy. Including them in the loss could harm the training.



(a) (b)

Fig. 2. (a) Some sample nuclei from the two datasets. (b) The F_1 scores of each class in the MoNuseg data obtained using 25% training data.

3 Experiments

To evaluate our method, we conduct experiments on two datasets, including the MoNuseg dataset [9] and our own Ki-67 nucleus dataset. In the MoNuseg dataset, there are four types of nucleus, (*i.e.*, Epithelial, Inflammatory, Fibroblast, Miscellaneous). In the Ki-67 dataset, there are also four types of nucleus, including immunopositive (non-)tumor nucleus and immunonegative (non-)tumor nucleus. The MoNuseg dataset contains 22462 nuclei and the Ki-67 dataset contains 17516 nuclei. For both datasets, 80% nuclei images are used for training, 20% of the training data is used for validation, and the rest are used for testing. A few samples of each type of nucleus are shown in Fig. 2(a).

We compare our method against two state-of-the-art SSDL methods, *i.e.,*TE [5] and MT [10], and a baseline fully supervised training method using the labeled data only. For each comparison, we train the different methods using only x% ($x = \{5, 10, 25, 50\}$ for the MoNuseg dataset, and $x = \{1, 5, 10\}$ for the Ki-67 dataset) of the training data as labeled data and the rest as unlabeled data. In fully supervised setting, the same network is trained using the labeled data only. Additionally, since the MoNuseg dataset is a publicly available dataset, we also show two results reported in [9], *i.e.,* CNN-SSPP and CNN-NEP. These two methods are fully supervised methods. In all the comparisons, weighted average F_1 score is used as evaluation metric. For the semi-supervised settings, and we report the average F_1 scores and their standard deviations of 5 runs on the testing data. In each of the 5 runs, a different set of labeled data are randomly selected.

3.1 Implementation Details

Network Architecture. In this paper, we adopt a network similar to the one used in [10]. The difference is the kernel size of the last two convolutional layers are set to 3. The input noise layer, ZCA layer, mean-only batch normalization are omitted. The advantage of our choice is that every component in our network can be implemented using standard Pytorch functions and scikit-learn package.

Table 1. Hyperparameter selection.

Hyperparameters	Value	Hyperparameters	Value
LP number of neighbors	7–10	Margin m in Eq. (3)	1.0
LP α	0.7	λ_{g1}	1.0
Initial learning rate	0.001	λ_{g2}	0.5
Weight decay	$2e$–4	Random translation (η, η')	2
λ_{EMA}	40	Random rotation (η, η')	Yes
Minibatch size	100	# of labeled samples per minibatch	31
LP iteration k	30	# of classes C	4

The features used in label propagation and Siamese loss are extracted from the intermediate layer (Fig. 1). Such design is chosen empirically. LP is conducted for each minibatch to build a connection graph. The Siamese loss based on the graph is computed in two terms Eq. (4). Specifically, the summation on L_{s1} is computed on 30 labeled data, and the summation on L_{s2} is from this 30 labeled data and 30 randomly selected unlabeled data. The coefficients for these two terms are shown in Table 1. The time complexity of LP is $k\mathcal{O}(CN^2)$, where k denotes the number of iterations, and C denotes the number of classes, and N denotes the minibatch size. With such overhead, our model can still be trained within 6 h on a GTX 1080 Ti GPU.

Hyperparameter Selection. Mostly we follow the parameter settings used in MT method [10]. The learning rate and the ramp function $w(\tau)$ are ramped up and down during the 150000 global steps. Specifically, they are ramped up in the first 40000 global steps, then kept constant for the following 85000 global steps, and finally decreased to 0 in the last 25000 global steps. We use $w(\tau) = e^{-5(1-\tau/150000)^2}$ as the ramp-up function and $w(\tau) = e^{-12.5(\tau/150000)^2}$ as the ramp-down function. The other parameter setting in our method are shown in Table 1.

3.2 Results and Analysis

Tables 2 and 3 illustrate that our method outperforms the state of the arts, especially when using less labeled data. For the MoNuseg dataset (Table 2), our method achieves around 2% higher F_1 scores compared to MT and TE methods when using 5% and 10% of the training data. Along with the increase of labeled data used, the performance of all the semi-supervised methods converges. In comparison with the baseline fully supervised method using labeled data only, our performance is higher by large margin. In contrast to the results reported in [9], our method outperforms CNN-SSPP using only 5% labeled data and achieves the performance close to CNN-NEP using only 25% labeled data. It is worth note that our method and CNN-SSPP take the nucleus patch as the sole

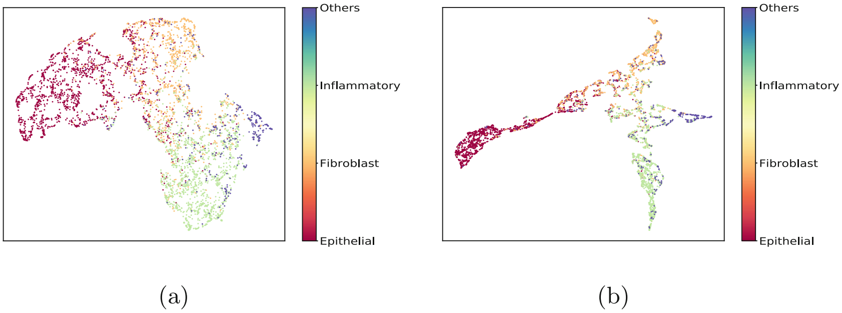


Fig. 3. Embeddings of the MoNuseg testing data projected to 2D space using UMAP [6]. (a) The feature embedding obtained by MT. (b) The embedding obtained by our method.

input while CNN-NEP takes into account the contextual information around the nucleus. This means CNN-NEP is actually using more labeled data. Moreover, the contextual information around the nucleus may not be a general approach for all nucleus classification problems. CNN-SSPP and CNN-NEP are both fully supervised methods. They are listed in the column 50% labels in Table 2, because they are based on two-fold cross validation [9]. Since the MoNuseg dataset is an imbalanced dataset, we show a comparison of the F_1 scores for each class in Fig. 2(b). Finally, to demonstrate the effect of the graph based clustering loss, we show the feature embedding of the MoNuseg testing data in Fig. 3.

For the Ki-67 dataset, we observed similar behavior. Our method outperforms the MT, TE and fully supervised method. Ablation studies are designed to show the effect of our proposed graph based clustering loss. Since the graph based clustering loss consists of two parts: (i) L_{s1} computed on the labeled data only; and (ii) L_{s2} computed between the unlabeled data and the labeled data. We train our model with one of the two losses removed and show the performance in Table 3. It can be seen that the performance drops if either one of them is removed. This shows the advantage of learning from a graph constructed on both labeled and unlabeled data.

Table 2. F_1 ± std over 5 runs on MoNuseg dataset [9].

Supervised methods	5% labels	10% labels	25% labels	50% labels	All labels
Labeled data only	63.21 ± 1.92	64.97 ± 1.72	73.04 ± 0.54	74.5 ± 0.72	78.15 ± 0.25
CNN-SSPP [9]	-	-	-	74.8	-
CNN-NEP [9]	-	-	-	78.4	-
Semi-supervised	**5% labels**	**10% labels**	**25% labels**	**50% labels**	**All labels**
TE [5]	73.2 ± 0.51	74.01 ± 0.85	76.46 ± 0.24	76.48 ± 0.21	76.57 ± 0.26
MT [10]	73.07 ± 0.56	74.35 ± 0.54	76.42 ± 0.56	76.59 ± 0.33	78.1 ± 0.29
Ours	**75.02** ± 0.55	**75.79** ± 0.23	**76.72** ± 0.17	**76.89** ± 0.25	**78.3** ± 0.23

Table 3. F_1 ± std over 5 runs on Ki-67 dataset.

Methods	1% labels	5% labels	10% labels	All labels
Labeled data only	-	75.99 ± 3.05	75.87 ± 1.02	79.06 ± 0.37
TE [5]	72.62 ± 4.47	76.02 ± 2.34	78.25 ± 0.48	79.22 ± 0.49
MT [10]	72.69 ± 4.7	76.92 ± 2.02	78.69 ± 0.47	79.41 ± 0.52
Ours	$\mathbf{74.9} \pm 3.41$	$\mathbf{79.32} \pm 0.73$	$\mathbf{79.79} \pm 0.59$	$\mathbf{79.91} \pm 0.39$
Ours w/$\lambda_{g1} = 0$	73.46 ± 3.59	77.72 ± 1.65	77.95 ± 0.59	-
Ours w/$\lambda_{g2} = 0$	73.2 ± 3.31	77.84 ± 1.27	78.15 ± 0.46	79.19 ± 0.83

4 Conclusion

In this paper, we presented a novel semi-supervised deep learning method for nucleus classification. The proposed method is a type of self-ensembling based deep learning methods with additional regularization from the local and global consistency criteria. The consistencies enable the framework to learn a better distance metric such that the resultant model outperforms the state-of-the-art self-ensembling methods on two nucleus classification datasets. The proposed approach is general for image classification, thus can be easily adapted for many other image classification tasks.

References

1. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a “Siamese” time delay neural network. In: Advances in Neural Information Processing Systems (NIPS), pp. 737–744 (1994)
2. Chapelle, O., Weston, J., Schölkopf, B.: Cluster kernels for semi-supervised learning. In: Advances in Neural Information Processing Systems (NIPS), pp. 601–608 (2003)
3. Chen, H., Dou, Q., Wang, X., Qin, J., Heng, P.A., et al.: Mitosis detection in breast cancer histology images via deep cascaded networks. In: AAAI, pp. 1160–1166 (2016)
4. Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 411–418. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40763-5_51
5. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
6. McInnes, L., Healy, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
7. Miyato, T., Maeda, S.I., Ishii, S., Koyama, M.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) (2018)

8. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 3546–3554 (2015)
9. Sirinukunwattana, K., Raza, S.E.A., Tsang, Y.W., Snead, D.R., Cree, I.A., Rajpoot, N.M.: Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. IEEE Trans. Med. Imaging (TMI) **35**(5), 1196–1206 (2016)
10. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems (NIPS), pp. 1195–1204 (2017)
11. Valpola, H.: From neural PCA to deep unsupervised learning. In: Advances in Independent Component Analysis and Learning Machines, pp. 143–171. Elsevier (2015)
12. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 639–655. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_34
13. Xing, F., Su, H., Neltner, J., Yang, L.: Automatic Ki-67 counting using robust cell detection and online dictionary learning. IEEE Trans. Biomed. Eng. (TBME) **61**(3), 859–870 (2014)
14. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems (NIPS), pp. 321–328 (2004)