Contrastive Learning and Iterative Meta-Pseudo-Labeling on 2D Projections for Deep Semi-Supervised Learning

David Aparco-Cardenas, Jancarlo F. Gomes, Alexandre X. Falcão and Pedro J. de Rezende
Institute of Computing (IC), University of Campinas (UNICAMP), Campinas, Brazil
{david.cardenas,jgomes,rezende,afalcao}@ic.unicamp.br

Abstract—**The scarcity of accurately labeled data critically hampers the usage of deep learning models. This issue is highlighted in areas (e.g., biological sciences) where data annotation results in an expert-demanding, labor-intensive and error-prone task. While state-of-the-art semi-supervised approaches have proven effective in circumventing this limitation, their reliance on pre-trained architectures and large validation sets to deliver effective solutions still poses a challenge. In this work we introduce an iterative contrastive-based meta-pseudo-labeling method for training non-pre-trained custom CNN architectures for image classification in conditions of limited labeled and abundant unlabeled data, with no dependency on a validation set. It generates multiple models across a few iterations, which are in turn exploited in an ensemble manner to label the unlabeled data and train a final classifier. Our approach starts by capitalizing on contrastive learning to enhance the representation ability of two collaborative networks while eliminating the need of pre-trained architectures. Then, during each iteration, the networks are trained within a teacher-student based cross-training setup, where OPFSemi (teacher) propagates labels from labeled to unlabeled on the non-linear 2D latent space projections of each network's (student) deep features; afterward, the pseudo-labels with the highest top 10% confidence, per class, are picked to fine-tune the other network in a cross-training manner, jointly mitigating confirmation bias and overfitting while improving the generalization ability of the networks as iterations evolve. Our method is evaluated on three challenging biological image datasets with only 5% of labeled samples, demonstrating its effectiveness and robustness when compared to two direct baselines and six state-of-the-art methods from three different semi-supervised learning paradigms.**

I. Introduction

In recent years, deep learning techniques have achieved remarkable success for their versatility in addressing a broad range of problems across a wide spectrum of areas and fields, such as computer vision, natural language processing, and speech recognition [1]. This success can mainly be attributed to the abundance of data available, and the concurrent advancements in both deep learning algorithms and computing power. However, the available data is most often unlabeled in real-life scenarios, posing a significant challenge for fully-supervised methods that heavily rely on labeled data. In this context, *semi-supervised learning* (SSL) methods aim to mitigate the scarcity of labeled samples by leveraging unlabeled data to enhance the generalization ability of the predictive model. Specifically, within the realm of deep neural networks, this approach is called *deep semi-supervised learning* (DSSL). Over the years, a multitude of DSSL methods employing diverse strategies have been proposed [2]. Among these, *pseudo-labeling* methods stand out for their prominence due to their straightforward yet effective training mechanism consisting in inferring pseudo-labels for the unlabeled samples based on the model's highest confidence predictions, which are then used to regularize and improve the model during training.

Recently, Benato *et al.* [3], [4] introduced a meta-pseudo-labeling approach known as *confidence Deep Feature Annotation* (conf-DeepFA) for semi-supervised training of CNNs, following a teacher-student approach. This method builds on the *Deep Feature Annotation* (DeepFA) technique [5] by iteratively exploiting label propagation via OPFSemi [6] (teacher) on the non-linear 2D projection of the deep features of a pre-trained CNN architecture (student). It thus far has demonstrated high accuracy on various datasets while requiring minimal labeled samples (*e.g.*, ∼1% of the dataset), without relying on a validation set. However, its application using non-pre-trained custom CNN architectures remains unexplored. Moreover, a potential challenge of the method is confirmation bias, where incorrect high-confidence pseudo-labeled samples may adversely affect model regularization [7].

More recently, Wang *et al.* [8] introduced an iterative method based on contrastive learning and self-training pseudo-labeling to address the classification of remote sensing images with limited labeled training data. Their method implements two independent yet synergic CNNs working in a cross-training procedure for robust generalization capability. It not only leverages contrastive learning to enhance the representation ability of the networks during the initial step, but also integrates self-training pseudo-labeling effectively to harness unlabeled data, making it particularly suited for scenarios where labeled data is scarce.

Herein, we present an iterative contrastive-based meta-pseudo-labeling method that expands on the DeepFA methodology for training non-pre-trained custom CNN architectures under conditions of limited labeled and plentiful unlabeled data. It implements two collaborative CNNs adopting a cross-training strategy to enhance the accuracy of pseudo-label generation and mitigate confirmation bias. This iterative approach integrates contrastive learning with meta-pseudo-labeling to effectively utilize unlabeled data, resulting in improved performance of custom CNN architectures under resource-constrained conditions.

Unlike the approach by Wang *et al.* [8], our method uses OPFSemi as teacher to generate pseudo-labels through label propagation on the 2D projection of the network's deep features. Instead of using an ensemble of the networks produced across all iterations for prediction on the test set, our method capitalizes on them to label the unlabeled data, which is then used to train a final CNN model. On the other hand, in contrast to conf-DeepFA, we leverage on contrastive learning to initialize the weights of non-pre-trained custom CNN architectures using only a reduced set of labeled samples, eliminating the need for pre-trained architectures. Moreover, our method implements two synergic CNNs that minimize an iterative categorical cross-entropy (ICE) loss function within a cross-training framework, aiming to obtain more reliable accurate pseudo-labels and mitigating confirmation bias. The final CNN model produced by our method is evaluated on three challenging biological image datasets, benchmarked against its two direct baselines and six other state-of-the-art Deep Semi-Supervised Learning (DSSL) approaches.

The main contributions of this study are:

1) We introduce a novel iterative contrastive-based meta-pseudo-labeling approach that builds on DeepFA to train non-pre-trained custom CNN architectures for image classification under conditions of limited labeled and abundant unlabeled data.
2) We implement DeepFA within a cross-training strategy using two CNNs that collaborate to minimize an ICE loss function. This integration aims to mitigate confirmation bias, produce more accurate pseudo-labels, and enhance the generalization capability of the final model.

The remainder of this paper is organized as follows. Section II presents out proposed methods in detail, including the iterative contrastive-based meta-pseudo-labeling approach and the cross-training strategy with two synergic CNNs. Next, Section III describes the conducted experiments and discusses the results. Lastly, Section IV concludes the paper with insights and future work directions.

II. Proposed Approach

In a Semi-Supervised Learning (SSL) framework, a training set $\mathcal{Z} = \{x | x \in X\}$ consists of two disjoint subsets: a labeled set $\mathcal{Z}_L = \{(x, y) | x \in X, y \in Y\}$ and an unlabeled set $\mathcal{Z}_U = \{x | x \in X\}$, where $\mathcal{Z} = \mathcal{Z}_L \cup \mathcal{Z}_U$, $\mathcal{Z}_L \cap \mathcal{Z}_U = \emptyset$, and $|\mathcal{Z}_L| \ll |\mathcal{Z}_U|$. Here, x represents a sample, and y denotes its associated label. We operate under the assumption that samples in both $\mathcal{Z}_L$ and $\mathcal{Z}_U$ originate from the same underlying distribution.

A. Network architecture

Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be the networks used during the contrastive learning stage that share the same architecture. After adding a decision layer to each network, we obtain networks $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ used for the fine-tuning stage. The architecture of both $\mathcal{N}_1$ and $\mathcal{N}_2$ comprises a sequence of four convolutional layers with 64, 128, 256, and 512 filters each as *encoder* and two dense (fully-connected) layers as *decoder*. Each convolutional layer of the encoder consists of a filter bank, ReLU activation, max-pooling with stride 2 and batch normalization. The encoder comprises two dense layers, with 512 and 256 neurons each, followed by ReLU activation. This is illustrated in Figure 2.

B. Overview of the Proposed Approach

The overall procedure of the proposed method can be divided into three successive main stages: network initialization by contrastive learning, iterative meta-pseudo-labeling and final model training. The first step consists in independently initializing – by means of contrastive learning – the weights of $\mathcal{N}_1$ and $\mathcal{N}_2$. The training set comprises pairs of images from $\mathcal{Z}_L$, while each network adopts a siamese structure implementing two weight-sharing sub-networks (see Figure 2). The second step starts by fine-tuning the networks in a cross-training fashion as follows: in subsequent iterations (excluding the first where only $\mathcal{Z}_L$ is used), a decision layer is added to $\mathcal{N}_{1/2}$ transferring the weights learned in the first step. Next, $\mathcal{N}_{1/2}^*$ is fine-tuned by minimizing an ICE loss on the set comprising $\mathcal{Z}_L$ and the pseudo-labels selected in the previous iteration from deep feature annotation on $\mathcal{N}_{2/1}^*$. Once fine-tuning is finished, the last dense layer's latent space of $\mathcal{N}_{1/2}^*$ is projected onto 2D for downstream label propagation via OPFSemi. The pseudo-labeled samples with the highest top 10% confidence, per class, are chosen to fine-tune $\mathcal{N}_{2/1}^*$ in the next iteration.

The aforesaid procedure is repeated for a fixed number of iterations. Lastly, the third step involves training a final model $\mathcal{N}_F$ of the same architecture as $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$. Let T be the chosen number of iterations, we use the probability vectors yielded by the $2T$ models produced across iterations to label the entire unlabeled set $\mathcal{Z}_U$ producing $\mathcal{Z}_U^L$. Once labeled, we train $\mathcal{N}_F$ on the set $\mathcal{Z}_F = \mathcal{Z}_L \cup \mathcal{Z}_U^L$ and return it as the final classifier. Figure 1 illustrates the two first steps of the proposed approach.

C. Network Initialization by Contrastive Learning

We adopt a contrastive learning as pre-training step as a way of enhancing the representation ability of the networks. In this respect, contrastive learning acts as a dimensionality reduction procedure by contrastively mapping a set of high-dimensional input data points (images) to lower-dimensional representations, encouraging the representations of semantically similar pairs to be close, and those of dissimilar pairs to be distant from each other in the lower-dimensional manifold.

During the learning process, each network adopts a siamese structure modeled as two weight-sharing sub-networks (see Figure 2). Let $x_1, x_2 \in \mathcal{Z}_L$ be a pair of input images. In the course of training, augmented versions of x_1 and x_2 are fed to different branches of the Siamese network. The distance function $\mathcal{D}$ between the lower-dimensional representations of x_1 and x_2 generated by a network $\mathcal{N}$ is defined as the Euclidean distance as $\mathcal{D}(x_1, x_2) = \|\mathcal{N}(x_1) - \mathcal{N}(x_2)\|$.

Let y_t be a binary label assigned to the pair x_1, x_2, where $y_t = 0$ if they are deemed similar and $y_t = 1$ if they are deemed dissimilar. In order to balance the large

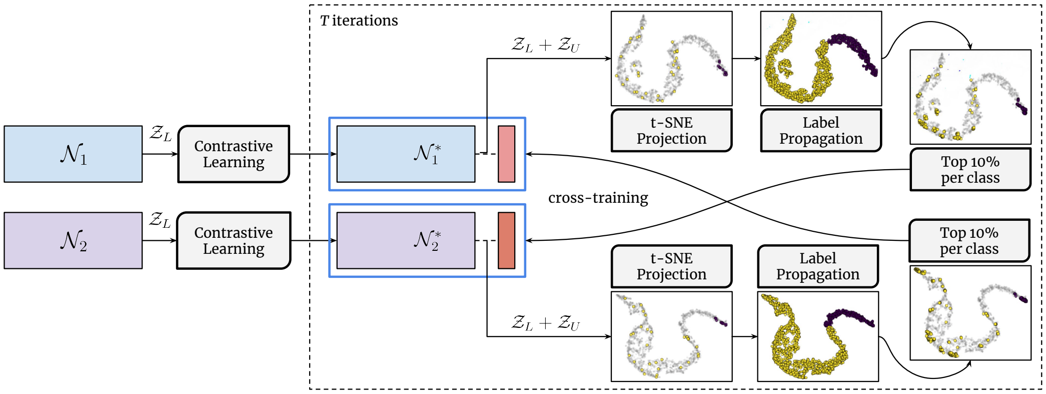


Fig. 1. Network initialization by contrastive learning and iterative meta-pseudo-labeling with a cross training strategy.

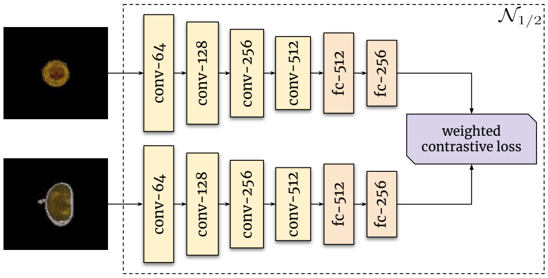


Fig. 2. Weight-sharing Siamese structure during contrastive learning.

difference between the number of similar and dissimilar pairs, a weight factor τ is introduced into the loss function. The weighted contrastive loss function is $\mathcal{L} = \sum_{i=1}^{P} L(p^i)$ with $L(p^i) = \tau(1-y_t)\frac{1}{2}(\mathcal{D}^i)^2 + (y_t)\frac{1}{2}\{\max(0, m-\mathcal{D}^i)\}^2$, where $p^i = (y_t, x_1, x_2)^i$ is the i-th labeled sample pair and P is the number of training pairs. The margin m is empirically set to 2. Let n be the number of classes and let k be the number of samples per class for a given balanced dataset. Then, the ratio between the number of similar and dissimilar pairs is $\frac{n\binom{k}{2}}{\binom{nk}{2}-n\binom{k}{2}} = \frac{k-1}{k(n-1)} \approx \frac{1}{n-1}$. In this way, the value of τ is set to $n-1$ to balance the contribution of similar pairs in the loss function as is used in [8].

D. Label Propagation by OPFSemi

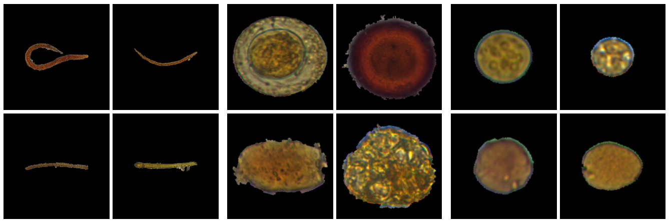
The pseudo-labeling procedure relies on label propagation by OPFSemi [6], a graph-based algorithm based on the optimum-path forest (OPF) methodology that has already been utilized for pseudo-labeling in recent works [3]–[5]. Let $\mathcal{Z}^\beta \subset \mathrm{R}^2$ be the non-linear projection on 2D of the last dense layer's deep features of $\mathcal{Z}$. For each network, $\mathcal{Z}^\beta$ is computed by forward passing $\mathcal{Z}$ through the network and projecting the latent space of the last dense layer onto 2D by means of t-SNE. OPFSemi is executed on the graph induced by the projection set $\mathcal{Z}^\beta$. The algorithm transforms $\mathcal{Z}^\beta$ into a complete graph where samples are encoded as nodes. The subset of labeled samples $\mathcal{Z}_L^\beta$ becomes the set of prototypes and their labels are propagated to their most closely connected unlabeled samples in $\mathcal{Z}_U^\beta$, partitioning the graph into an optimum-path forest rooted at $\mathcal{Z}_L^\beta$. Let $\lambda'(u)$ be the pseudo-label assigned to $u \in \mathcal{Z}_U^\beta$ by OPFSemi, and let $\lambda(s)$ be the label of $s \in \mathcal{Z}_L^\beta$, such that s is the root of the optimum-path tree to which u is connected – *i.e.*, $\lambda'(u) = \lambda(s)$. Let $c(u)$ be the cost of the optimum-path offered from s to u, and let c' be the cost of the second least optimum-path offered to u from $t \in \mathcal{Z}_L^\beta$, where $s \neq t$ and $c' > c(u)$. The labeling confidence value $v(u) = \frac{c'}{c(u)+c'} \in [0,1]$ is computed and assigned to each $u \in \mathcal{Z}_U^\beta$.

After label propagation, the set $\mathcal{Z}^\gamma$ is constructed comprising $\mathcal{Z}_L$ and the pseudo-labeled samples with the highest 10% confidence, per class. In this way, we guarantee the selection of reliable pseudo-labeled samples from all classes in contrast to using a global labeling confidence threshold, which may overlook pseudo-labeled samples from classes where the mean labeling confidence value is lower than for other classes.

E. Iterative Cross-Training and Final Model Training

During the first iteration, we fine-tune both networks on the set $\mathcal{Z}_L$, while from the second iteration onwards we fine-tune $\mathcal{N}_1^*$ on the set $\mathcal{Z}^{\gamma(2)}$ obtained from deep feature annotation on $\mathcal{N}_2^*$ in the previous iteration, and vice versa for $\mathcal{N}_2^*$.

In regular conditions, the selected pseudo-labeled samples are used to retrain the original network in the subsequent iteration. However, this may cause overfitting and confirmation bias, thereby hindering the network's generalization ability. Therefore, the idea of using pseudo-labeled samples generated by another network comes from the fact that a network improves its generalization capability when trained on new data. Moreover, pseudo-labeled samples are expected to become more reliable in later iterations than in earlier ones. For this reason, during the optimization phase we adopt an iterative categorical cross-entropy (ICE) loss function that regulates the contribution of pseudo-labeled samples as iterations proceed. The ICE loss function is defined as $\mathrm{ICE}(y_p, y, t, T, f) = f(1-\frac{t-1}{2T})\,\mathrm{CE}(y_p, y) + (1-f)\frac{t-1}{2T}\,\mathrm{CE}(y_p, y)$, where y_p and y are the network's outputs and training labels, T is the total number of iterations, and $t \in \{1, \ldots, T\}$ is the iteration counter. CE stands for the classical cross-entropy loss function, while f is a binary flag that indicates whether the input image



(a) *Helminth larvae* (b) *Helminth eggs* (c) *Protozoan cysts*

Fig. 3. Examples of images from *Helminth larvae*, *Helminth eggs* and *Protozoan cysts*. The first row shows examples of parasites, while the second row shows examples of impurities.

is labeled ($f = 1$) or pseudo-labeled ($f = 0$). During the first iteration, the ICE loss function degenerates into CE. As iterations evolve, the contribution of pseudo-labeled samples in the loss function gradually increases, while that of labeled samples decreases, nonetheless preserving its dominance.

Once the iterative process is complete, we leverage on the $2T$ models produced across T iterations (2 per iteration) to label the unlabeled set $\mathcal{Z}_U$. Let $p\vec{v}_t^{(i)}$ be the prediction probability vector of the network i during iteration t. For each unlabeled sample $u \in \mathcal{Z}_U$, we sum the average prediction probability vectors of the two models produced during each iteration, then the label of u, denoted by $\lambda(u)$, is determined as the index of the highest value of the resultant vector as follows $\lambda(u) = \text{argmax}\left\{\frac{p\vec{v}_1^{(1)}+p\vec{v}_1^{(2)}}{2} + \ldots + \frac{p\vec{v}_T^{(1)}+p\vec{v}_T^{(2)}}{2}\right\}$. Lastly, we train from scratch network $\mathcal{N}_F$, of the same architecture as $\mathcal{N}_1$ and $\mathcal{N}_2$, on the set $\mathcal{Z}_f = \mathcal{Z}_L \cup \mathcal{Z}_U^L$, where $\mathcal{Z}_U^L$ is the recently labeled-unlabeled set.

III. Experiments and Results

A. Datasets

We employ three challenging datasets from a parasite biological image collection [9] to evaluate the performance of our method. The collection includes the most common human intestinal parasites found in Brazil, and prevalent in most countries with tropical, subtropical, and equatorial climates and responsible for significant public health problems. The datasets consist of optical microscopy images rescaled to $200 \times 200 \times 3$ pixels. They comprise the following datasets *i*) *Helminth larvae*; *ii*) *Helminth eggs*; and *iii*) *Protozoan cysts*. It is noteworthy that the inherent imbalance in the datasets and the close resemblance between impurities and parasites pose an added challenge to the classification problem. A detailed description of the datasets is given in Table I, while some image examples of parasites and impurities for (*i*)–(*iii*) are shown in Figure 3.

B. Dataset Preparation

We use stratified three-fold cross-validation to split the dataset $\mathcal{Z}_D$ into $\frac{2}{3}$ for the training set $\mathcal{Z}$ and $\frac{1}{3}$ for the test set $\mathcal{Z}_T$ ($\mathcal{Z}_D = \mathcal{Z} \cup \mathcal{Z}_T$). The inclusion of a validation set would induce the need for more labeled data defeating the purpose of this work, therefore it is discarded. Moreover, during training, the input images are encoded using the Lab color space due to its contrast enhancement ability. In order to emulate the conditions of labeled data scarcity, we partition the training set for each split into labeled $\mathcal{Z}_L$ and unlabeled $\mathcal{Z}_U$ sets ($\mathcal{Z} = \mathcal{Z}_L \cup \mathcal{Z}_U$), such that $|\mathcal{Z}_L| = 5\% \times |\mathcal{Z}_D|$ and $|\mathcal{Z}_U| = 61.66\% \times |\mathcal{Z}_D|$. The labeled samples are selected from $\mathcal{Z}$ in a random stratified way for each split. Table II specifies the number of samples in both $\mathcal{Z}_L$ and $\mathcal{Z}_U$ for each dataset.

TABLE I
Description of the datasets.

Dataset	Number	Category	Class ID
Helminth larvae	446	*Strongyloides stercoralis*	1
	3 068	Impurities	2
	3 514	**Total**	
Helminth eggs	348	*Hymenolepis nana*	1
	80	*Hymenolepis diminuta*	2
	148	Ancylostomatidae	3
	122	*Enterobius vermicularis*	4
	337	*Ascaris lumbricoides*	5
	375	*Trichuris trichiura*	6
	122	*Schistosoma mansoni*	7
	236	*Taenia* spp.	8
	3 344	Impurities	9
	5 112	**Total**	
Protozoan cysts	719	*Entamoeba coli*	1
	78	*Entamoeba histlytica / E. dispar*	2
	724	*Endolimax nana*	3
	641	*Giardia duodenalis*	4
	1 501	*Iodamoeba bütschlii*	5
	189	*Blastocystis hominis*	6
	5 716	Impurities	7
	9 568	**Total**	

TABLE II
Number of samples in the $\mathcal{Z}_L$ and $\mathcal{Z}_U$ subsets of each dataset.

	H. larvae	*H. eggs*	*P. cysts*
$\lvert\mathcal{Z}_L\rvert$	177	274	488
$\lvert\mathcal{Z}_U\rvert$	2167	3138	5893

C. Experimental Setup

All the experiments in this section were implemented in Python 3.9.12 using PyTorch 2.0.1. For t-SNE, we used the implementation available in *sci-kit learn* 1.4.2, keeping its default parameters. It is worth noting that OPFSemi is parameter-free. For the contrastive learning stage, we use Adam optimizer with batch size 32, learning rate 10^{-4} and number of epochs 100. For the iterative cross-training optimization stage, we use stochastic gradient descent (SGD) as optimizer with batch size 32, momentum 0.9, weight decay 10^{-3}, Nesterov momentum, and number of epochs 120. The SGD optimizer adopts a polynomial learning rate decay with power 1.0. The number of iterations for the iterative process was empirically obtained and set to $T = 7$.

For the sake of comparison fairness, in all experiments all methods adopted the same architecture as described in Section II-A. Also, the same three-fold cross-validation splits detailed in Section III-B were used to evaluate the methods. The metrics used to assess the performance of the methods are accuracy and Cohen's κ. The latter is used to obtain a more reliable measure than accuracy since we are addressing unbalanced datasets. Cohen's κ, hereafter denoted by κ, measures the degree of agreement between the classifier's prediction and the ground-truth, where $\kappa \in [-1, 1]$. A value of $\kappa = 1$ indicates total agreement, while $\kappa \leq 0$ indicates a lower chance of agreement. The mean and standard deviation across splits of each metric are reported for each method.

D. *Comparison with baselines*

The baselines for our comparative analysis are the methods by Wang *et al.* [8] and conf-DeepFA [4]. This experiment seeks the validate our method against these baselines, from which some components are borrowed. It is worth pointing out that, as with our method, the baselines do not require a validation set. Since conf-DeepfA relies on a pre-trained encoder, we use contrastive learning to initialize the encoder's weights. For conf-DeepfA, we set the global labeling confidence threshold as $\tau = 0.8$. Table III shows the mean and standard deviation of accuracy and κ across splits for all three methods. It can be seen that our approach outperforms its counterparts in both metrics for all datasets.

TABLE III
TEST RESULTS OF ACCURACY AND κ FOR BASELINES.

Method	**Metric**	**Datasets**		
		Helminth larvae	*Helminth eggs*	*Protozoan cysts*
Wang *et al.* [8]	accuracy	0.970 ± 0.022	0.906 ± 0.004	0.886 ± 0.013
	κ	0.878 ± 0.078	0.822 ± 0.006	0.806 ± 0.021
conf-DeepFA [4]	accuracy	0.966 ± 0.016	0.884 ± 0.010	0.799 ± 0.060
	κ	0.867 ± 0.064	0.785 ± 0.024	0.663 ± 0.086
Proposed	accuracy	**0.976 ± 0.015**	**0.946 ± 0.005**	**0.922 ± 0.009**
	κ	**0.889 ± 0.064**	**0.901 ± 0.009**	**0.870 ± 0.014**

1) Influence of contrastive learning: The results reveal contrastive learning as an effective initializer for network weights from a reduced set of labeled samples, thereby serving as a good alternative to circumvent the prerequisite of pre-trained models in DeepFA-based methods. Also, it proves to be effective in enabling the use of conf-DeepFA with non-pre-trained custom CNN architectures. In contrast to both conf-DeepFA and the method by Wang *et al.*, our method initializes the network weights up to the last dense layer, rather than only the encoder part, which enhances the representations learned by the network for subsequent projection and label propagation as demonstrated by the results.

2) Influence of layer selection for 2D projection: Our method projects the latent space of the last dense layer, instead of the last convolutional layer, during the iterative cross-training stage, which has a significant impact on the results. Figure 4 shows the projections of the latent space of the last dense and the last convolutional layers for the *Protozoan cysts* dataset, where the first provides a better separation among classes, which in turn favors label propagation by OPFSemi. The rationale behind this choice is based on the work by Rauber *et al.* [10], who demonstrated that the deeper the layer in an effectively trained network, the more separated the classes are likely to be in the latent space 2D projection.

3) Influence of iterative cross-training: The integration of DeepFA into the cross-training procedure using two collaborative networks, together with the adoption of the ICE loss, helps to mitigate the issues of confirmation bias and overfitting, enhancing the generalization ability of the networks as iterations proceed, which in turn is reflected in the generalization ability of the final model. This is further validated by the direct comparison between the method by Wang *et al.* and conf-DeepFA, where the latter outperforms the former in all datasets.

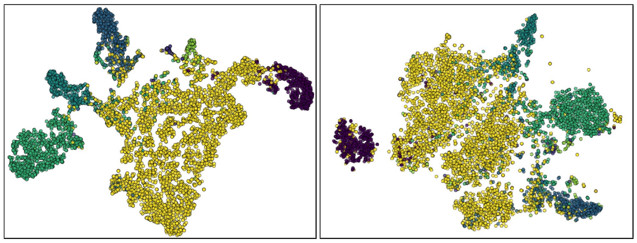

(a) Last dense layer (b) Last convolutional layer

Fig. 4. Projections of the latent space of the last dense (left) and the last convolutional layer (right) for the *Protozoan cysts* dataset.

4) Influence of OPFSemi as teacher: On the other hand, the incorporation of OPFSemi as teacher to generate pseudo-labels via label propagation significantly improves the results by Wang *et al.*, which performs pseudo-labeling in a self-training manner. This result demonstrates the effectiveness of OPFSemi as teacher by providing more reliable pseudo-labels in suitable projection conditions.

E. *Comparison with other state-of-the-art methods*

In this experiment, we compare our approach with other state-of-the-art (SOTA) DSSL methods. Six SOTA DSSL methods from three categories according to their solution strategy were selected: *Pseudo-labeling*: Pseudo-label [11]; *Consistency Regularization*: Π-model [12], Mean Teacher [13], VAT [14] and UDA [15]; and *Hybrid*: FixMatch [16]. We used the Pytorch implementation available in the Unified SSL Benchmark (USB) library [17]. We employed the initial labeled set $\mathcal{Z}_L$ as validation set, since all six DSSL methods rely on it to select the best-performing model. Table IV shows the mean and standard deviation of accuracy and κ for all methods. It can be seen that our method outperforms all its counterparts in both metrics for all three datasets.

Our method shows a clear superiority over Pseudo-label, suggesting that a meta-pseudo-labeling approach introduces enhancements over the standard self-training pseudo-labeling strategy, which validates the usage of OPFSemi as teacher. Consistency regularization approaches exhibit competitive results on all three datasets, hinting that including data augmentation-based strategies and consistency constraints in the loss function may further enhance the performance of our method. Moreover, Fixmatch, a hybrid method that capitalizes on both self-training-based pseudo-labeling and consistency regularization, shows competitive results on all datasets. Therefore, the incorporation of an auxiliar model as teacher (*e.g.*, OPFSemi) may improve the method's overall performance. A comparison between our method and Mean Teacher, another teacher-student-based approach, reveals the dominance

TABLE IV
TEST RESULTS OF ACCURACY AND κ FOR STATE-OF-THE-ART DSSL METHODS.

Method	**Metric**	**Datasets**		
		Helminth larvae	*Helminth eggs*	*Protozoan cysts*
Pseudo-label [11]	accuracy	0.960 ± 0.010	0.916 ± 0.008	0.903 ± 0.011
	κ	0.811 ± 0.057	0.850 ± 0.008	0.836 ± 0.023
Π-Model [12]	accuracy	0.962 ± 0.020	0.922 ± 0.003	0.912 ± 0.009
	κ	0.836 ± 0.081	0.855 ± 0.004	0.853 ± 0.016
Mean Teacher [13]	accuracy	0.969 ± 0.009	0.928 ± 0.017	0.906 ± 0.003
	κ	0.860 ± 0.037	0.870 ± 0.033	0.843 ± 0.005
VAT [14]	accuracy	0.965 ± 0.006	0.933 ± 0.013	0.912 ± 0.010
	κ	0.844 ± 0.031	0.880 ± 0.021	0.854 ± 0.018
UDA [15]	accuracy	0.966 ± 0.009	0.933 ± 0.002	0.914 ± 0.004
	κ	0.851 ± 0.033	0.880 ± 0.003	0.857 ± 0.007
FixMatch [16]	accuracy	0.968 ± 0.015	0.925 ± 0.011	0.914 ± 0.006
	κ	0.857 ± 0.060	0.862 ± 0.020	0.856 ± 0.011
Proposed	accuracy	**0.976 ± 0.015**	**0.946 ± 0.005**	**0.922 ± 0.009**
	κ	**0.889 ± 0.064**	**0.901 ± 0.009**	**0.870 ± 0.014**

of cross-training-based pseudo-labeling in a teacher-student setup. Our method shows highly competitive performance as compared to all consistency regularization-based counterparts, further validating the effectiveness of our strategy.

Another point worth noting is that the unbalanced nature of the dataset and the high similarity between some classes, in greater degree between impurities and parasites, can explain the variability found in the results from different methods. In this regard, our method shows its robustness and adaptability to new data providing more consistent results across all datasets.

IV. CONCLUSION

In this work, we introduced an iterative contrastive-based meta-pseudo-labeling method to train non-pre-trained custom CNN architectures for image classification in conditions of scarcity of labeled and abundance of unlabeled data. It does so by incorporating deep feature annotation (DeepFA) into an iterative cross-training procedure that implements two collaborative networks, which in turn minimizes an iterative categorical cross-entropy (ICE) loss that adjusts the contribution of pseudo-labels across iterations. This mitigates confirmation bias and overfitting by providing reliable pseudo-labels while improving the networks' generalization ability as iterations evolve. Also, the method capitalizes on contrastive learning to enhance the networks' representation ability and to circumvent the need for pre-trained architectures. We evaluated our method on three challenging biological image datasets emulating a labeled data scarcity scenario by labeling only 5% of the samples of each dataset. We compared our method to two direct baselines and to six other state-of-the-art DSSL from three different categories of approaches. For each dataset, our method improves its baselines and outperforms its state-of-the-art counterparts, demonstrating its effectiveness and robustness. As future work, we plan to further improve the results presented herein by incorporating data augmentation-based techniques and consistency constraints in the loss function with the intend of enhancing the accuracy of pseudo-labels and reducing the amount of labeled data.

ACKNOWLEDGMENT

The authors acknowledge grants from *Santander Bank Brazil*; CAPES/STIC-AMSUD, #88887.878869/2023-00; CNPq, #140930/2021-3, #314293/2023-0, #313329/2020-6, #442950/2023-3, #304711/2023-3; FAPESP, #2023/14427-8, #2023/04318-7, #2014/12236-1, #2013/07375-0.

REFERENCES

[1] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[2] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 8934–8954, 2023.

[3] B. C. Benato, A. C. Telea, and A. X. Falcão, "Iterative pseudo-labeling with deep feature annotation and confidence-based sampling," in *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2021, pp. 192–198.

[4] ——, "Deep feature annotation by iterative meta-pseudo-labeling on 2d projections," *Pattern Recognition*, vol. 141, p. 109649, 2023.

[5] B. C. Benato, J. F. Gomes, A. C. Telea, and A. X. Falcao, "Semi-automatic data annotation guided by feature space projection," *Pattern Recognition*, vol. 109, p. 107612, 2021.

[6] W. P. Amorim, A. X. Falcão, J. P. Papa, and M. H. Carvalho, "Improving semi-supervised learning through optimum connectivity," *Pattern Recognition*, vol. 60, pp. 72–85, 2016.

[7] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[8] C. Wang, H. Gu, and W. Su, "Sar image classification using contrastive learning and pseudo-labels with limited data," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[9] C. T. N. Suzuki, J. F. Gomes, A. X. Falcão, S. H. Shimizu, and J. P. Papa, "Automated diagnosis of human intestinal parasites using optical microscopy images," in *IEEE 10th International Symposium on Biomedical Imaging*, 2013, pp. 460–463.

[10] P. E. Rauber, A. X. Falcão, and A. C. Telea, "Projections as visual aids for classification system design," *Information Visualization*, vol. 17, no. 4, pp. 282–305, 2018.

[11] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, no. 2. Atlanta, 2013, p. 896.

[12] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations*, 2017.

[13] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[14] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[15] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[16] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[17] Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, R. Wang, L. Yang, Z. Zhou, L.-Z. Guo, H. Qi, Z. Wu, Y.-F. Li, S. Nakamura, W. Ye, M. Savvides, B. Raj, T. Shinozaki, B. Schiele, J. Wang, X. Xie, and Y. Zhang, "Usb: A unified semi-supervised learning benchmark for classification," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.