Pattern Recognition 109 (2021) 107612

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Semi-automatic data annotation guided by feature space projection

Bárbara C. Benato[a,*], Jancarlo F. Gomes[a], Alexandru C. Telea[b], Alexandre X. Falcão[a]

[a] Institute of Computing, University of Campinas, Campinas, Brazil
[b] Department of Information and Computing Sciences, Faculty of Science, Utrecht University, Utrecht, the Netherlands

ARTICLE INFO

Article history:
Received 2 July 2019
Revised 5 June 2020
Accepted 19 August 2020
Available online 22 August 2020

Keywords:
Semi-supervised learning
Unsupervised feature learning
Interactive data annotation
Autoencoder-neural networks
Data visualization

ABSTRACT

Data annotation using visual inspection (supervision) of each training sample can be laborious. Interactive solutions alleviate this by helping experts propagate labels from a few supervised samples to unlabeled ones based solely on the visual analysis of their feature space projection (with no further sample supervision). We present a semi-automatic data annotation approach based on suitable feature space projection and semi-supervised label estimation. We validate our method on the popular MNIST dataset and on images of human intestinal parasites with and without fecal impurities, a large and diverse dataset that makes classification very hard. We evaluate two approaches for semi-supervised learning from the latent and projection spaces, to choose the one that best reduces user annotation effort and also increases classification accuracy on unseen data. Our results demonstrate the added-value of visual analytics tools that combine complementary abilities of humans and machines for more effective machine learning.

1. Introduction

Machine Learning (ML) models have been extensively investigated and used for regression and classification problems [1–3]. More recently, Convolutional Neural Networks (CNNs) have shown great success in many applications, such as image/text classification [4] and speech recognition [5], since they require considerably less effort to optimize parameters than the common feature extraction pipeline [4]. However, CNNs may require a high number of labeled samples (annotated objects) for training [6].

While small labeled training sets can impair the ability of an ML model to correctly classify new samples (a problem known as *over-fitting* [7]), large unlabeled sets make visual inspection and annotation very expensive for the expert. Human costs become even so more prohibitive in domains that require specialized knowledge about the objects, like Medicine and Biology. Solutions for small labeled sets include data augmentation [8] and regularization methods [9]. For large unlabeled sets, semi-supervised classifiers have been used to propagate labels from a small supervised set to the many unsupervised samples by exploring the sample distribution in some feature space [10–12]. Yet, none of these approaches has combined the cognitive ability of humans in data abstraction with the ability of machines in data processing to increase the number of labeled objects.

Recent studies have investigated the use of feature space projections and visual analytics to understand and engineer ML models [13–17]. Such work addresses both aforementioned labeling cases with approaches for interactive data augmentation [15] and interactive data annotation [16,17] guided by feature space projections, respectively. Bernard et al. [16] have compared interactive data annotation in a feature space projection with an active learning technique, in which experts supervise and annotate samples selected by a classifier and the classifier is retrained to annotate and select more samples in the original feature space. They discovered that interactive data annotation in the feature space projection is superior to active learning. Benato et. al. [17] have showed that when the user propagates labels to a large unsupervised sample-set guided by the true-label knowledge of a few samples and by the visual information of the sample distribution in a feature space projection, the resulting labeled training-set is more correct than the one created by semi-supervised classifiers in the original feature space. Hence, classifiers trained from such interactively labeled sets can better predict labels of unseen test samples than those trained from automatically labeled sets. Yet, Bernard et al. [16] and Benato et al. [17] have not *combined* automatic and interactive approaches for label propagation — i.e, they have not been concerned with the user *effort* in visual data inspection and annotation.

In this work, we fill the above gap by proposing a semi-automatic approach that reduces user labeling effort while achieving better classification accuracy on unseen test sets. For this, we

* Corresponding author.
E-mail addresses: barbara.benato@ic.unicamp.br (B.C. Benato), jgomes@ic.unicamp.br (J.F. Gomes), a.c.telea@uu.nl (A.C. Telea), afalcao@ic.unicamp.br (A.X. Falcão).

https://doi.org/10.1016/j.patcog.2020.107612
0031-3203/

exploit the concept of *sample informativeness* from Active Learning (AL). Such approaches select samples for expert supervision based on their informativeness — i.e., potential to improve the design of a classifier from the knowledge of their true label [18], measured by the *confidence* of a classifier about the label assigned to a sample [19–22]. In our case, we propagate labels to samples with high-confidence values; and enable the expert focus on low-confidence values for manual label propagation. For this, the user visually analyzes the sample distribution in a 2D scatterplot created by the *t-Distributed Stochastic Neighbor Embedding* (*t*-SNE) technique [23], constructed similarly to [16,17], and the true-label knowledge of only a few samples per class. Although our method can explore further classifier improvement of the classifier by multiple iterations of AL with additional supervised samples, we solve data annotation from a single user interaction for label propagation with no sample supervision. For automatic label propagation, we evaluate two semi-supervised classifiers trained in both latent and projection spaces for automatic label estimation and choose the best one for our goal. We show that our semi-automatic label propagation (SALP) method achieves end-to-end better classification results as compared to both fully automatic label propagation and fully manual label propagation.

This work is organized as follows. Section 2 presents our semi-supervised data annotation approach. Section 3 presents the experimental setup, compared baselines, used datasets, and experimental results. Section 4 discusses our results. Section 5 concludes the paper.

2. Semi-automatic projection-based data annotation

Given a training set with a low number of supervised samples and a considerably larger number of unsupervised samples, our semi-automatic data annotation approach (Fig. 1) has four steps:

- *unsupervised feature learning:* We start by extracting features from the input dataset. To minimize the number of supervised samples needed, we adopt an unsupervised feature-learning procedure (Section 2.1);
- *feature space projection:* We create a feature space 2D projection that captures well the sample distribution in the latent feature space for further visual analysis;
- *semi-supervised label estimation:* We propagate labels automatically to high-confidence unlabeled samples, thereby increasing with training-set size with little effort and high quality (Section 2.3);
- *visual analysis:* The expert creates additional labeled samples to the above ones, by interactively propagating labels to the less-confident samples using the 2D projection (Section 2.4).

2.1. Unsupervised feature learning

We use an Autoencoder Neural Network (AuNN) [24,25] for unsupervised feature learning. AuNNs consist of two parts, encoder and decoder. The encoder maps the input samples to points in a reduced (latent) feature space; the decoder reconstructs these samples. The two parts are coupled and trained together by back-propagation. As cost function, we use the mean squared error between the original and reconstructed samples. For small errors, the obtained latent feature space is a reasonable representation of the original sample distribution. Hence, we train the AuNN with all labeled and unlabeled samples by ignoring labels. After evaluating several models, we decided for a Stacked Convolutional AuNN [24] — a neural network that presents convolutional layers and can usually obtain relevant latent features. For our experiments, we use image datasets. However, this latent feature learning can be used for any other kind of data that can be suitably mapped to the input layer of the encoder. Section 3 presents implementation details.

2.2. Feature space projection

Previous works indicate that 2D projections, created by the *t*-SNE algorithm [26,27], achieve this goal well [13,16,17], so we follow these (Section 2.2).

The dimension of the latent feature space can still be considered very high (with usually hundreds to thousands of features) and so unfeasible for visual inspection of the sample distribution. As previously mentioned, we wish to reduce the latent space to two dimensions by preserving as much as possible the relevant structure of the data. The most suitable techniques for this task seem to preserve local distances between samples and the *t*-SNE algorithm satisfies this criterion [23]. It is a non-linear projection that depends on the choice of two parameters: perplexity and number of iterations. Our choice for these parameters is discussed in Section 3.

2.3. Semi-Supervised label estimation

For semi-supervised label estimation, we consider two techniques that explore the sample distribution in a given feature

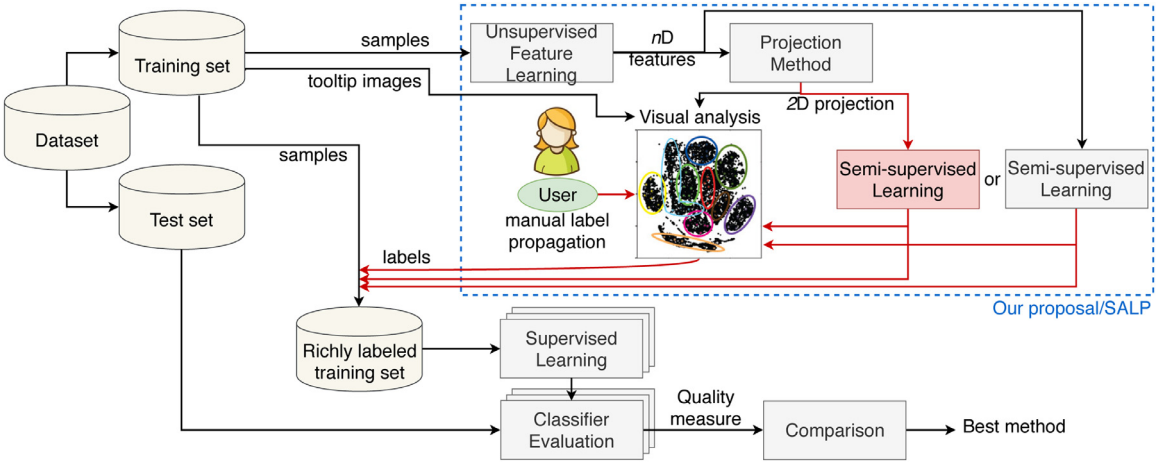

Fig. 1. Semi-automatic data annotation pipeline. We extract features by unsupervised learning from the training set and next use these to project this set to a 2D scatterplot. We next enrich the training set by propagating labels from supervised to unsupervised samples by automatic methods (in both latent and projection spaces) and by manual user-controlled methods. We finally compare the quality of the classifiers trained on such training sets to decide on the best label propagation method. Red indicates additions to earlier related work [17].

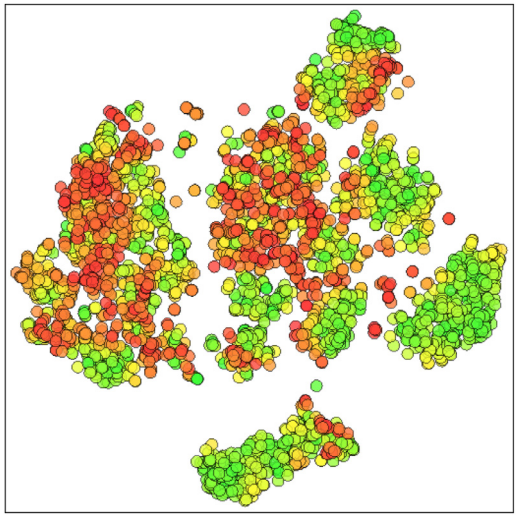

Fig. 2. Feature projection showing unsupervised samples from red (low confidence) to green (high confidence). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

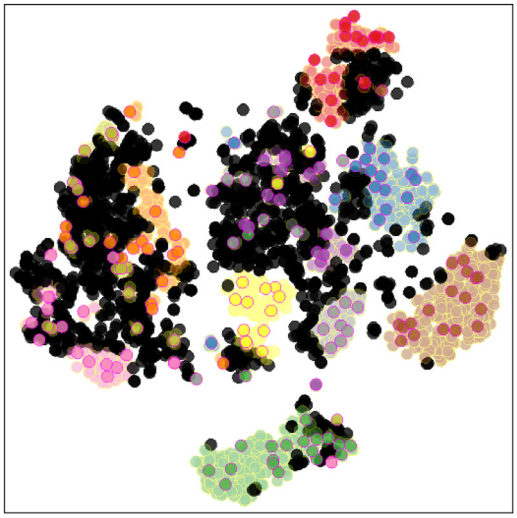

Fig. 3. Semi-automatic label propagation is done from the supervised samples (points colored by class, saturated colors, red border) first automatically to the unsupervised and high-confidence ones (light colors, no border). Remaining low-confident samples (black) are candidates for manual propagation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

space to propagate labels with confidence values from supervised to unsupervised ones: Laplacian Support Vector Machines (LapSVM) [28,29] and Semi-Supervised Classification by Optimum-Path Forest (OPF-Semi) [30]. We evaluate both methods on both latent and projection spaces. Given that the performance of OPF-Semi in label propagation is much higher than that of LapSVM (see Section 3), we select OPF-Semi to output confidence values, used next for our manual label propagation (Section 2.4). Additionally, we found that OPF-Semi in the projection space outperforms itself in the latent feature space (see Section 3). Hence, we use the 2D version of OPF-Semi for semi-automatic data annotation.

OPF-Semi maps (un)labeled samples to nodes of a graph and computes an optimum-path forest rooted at labeled samples. In this forest, each node s is conquered (labeled) by the root R that offers a path of minimum cost $k(R, s)$ to s. We use costs to compute label confidence values $c(s)$ as described in [20–22]. In brief: Let A and B be two roots for sample s so that A is the one that has conquered s ($k(R, s)$ is minimal) and B, having a different label than A, offers the second-best cost $k(B, s)$ to s. We assign the confidence $c(s) = 1 - k(A, s)/(k(A, s) + k(B, s))$, $c(s) \in [0, 1]$, to the label of s given by A. That is, if the second-best cost $k(B, s)$ is much larger than the minimal cost $k(A, s)$, the label A has a high confidence. We use the confidence as follows: All labels assigned by OPF-Semi having a confidence above a threshold τ are used as such in the training process. The threshold τ is chosen by the user based on the visual analysis of the feature projection with unsupervised samples colored by their confidence values from red (low c) to green (high c) (Fig. 2). Changing τ interactively by a slider lets the user (a) say that high-confidence samples can keep their likely good labels assigned by OPF-Semi and (b) focus on the remaining low-confidence samples to assign them labels by manual label propagation, described next in Section 2.4. Users can choose the exact threshold τ balancing how much they wish to trust OPF-Semi *vs* how many samples they are willing to label manually.

2.4. Manual label propagation

The added value of user-driven label propagation in a t-SNE projection was demonstrated by the interactive label propagation technique in [17] which we refer to next as ILP for brevity. However, ILP propagation is fundamentally affected by the quality of the latent features extracted by the AuNN (Section 2.1) *and* the quality of the t-SNE projection itself: If both these operations faithfully preserve the similarity of original samples, then the user can likely propagate labels well, by simply selecting points close in the projection to the supervised samples. If either the latent space or the projection create errors, which they inherently do [31], this will likely create wrong labels. We assist the user in this process as follows. We color the supervised points in the projection by their labels, and color all low-confidence unsupervised points s having $c(s) < \tau$ in black (Fig. 3). The black points are projected before the colored points, in order to minimize undesired occlusions. When moving the mouse pointer over a projected point, we show its sample image in a tooltip. The user next employs these three sources of information – proximity of unsupervised (black) points in the 2D projection to supervised (colored) ones, low-confidence value of the unsupervised points, and similarity of unsupervised-to-supervised tooltip images – to decide which unsupervised samples get which supervised label. Label propagation is next done simply by selecting desired points in the projection and clicking to assign them a supervised-point label.

3. Experiments and results

We next present the experimental setup, baselines, datasets, implementation details, and experimental results used for validating our semi-automatic data annotation method.

3.1. Experimental setup

We divide each available dataset D into three subsets for validation: a very small training set S with a few supervised samples per class (3%|D|); a considerably larger training set U with unsupervised samples for label propagation (67%|D|); and a set T with unseen test samples (30%|D|). Next, based on the user-chosen confidence threshold τ, we split U into high-confidence samples L_c, which get their label from OPF-Semi, and low-confidence ones L_i, which can be interactively labeled by the user. Note that $L_c \cap L_i = \emptyset$ and $L_c \cup L_i \neq U$, since the user can choose not to label L_i entirely, to minimize manual labeling effort. We randomly split D into S, U, and T this way three times and repeat the evaluation — i.e., label

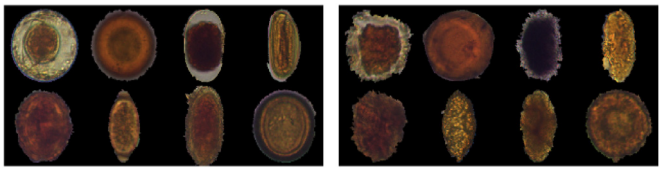

Fig. 4. Examples of each species of *H. Eggs* (left) and similar images of impurities (right).

Table 1
Number of samples in *S*, *U* and *T* for each dataset: MNIST, Parasites, and Parasites with impurity (I).

Dataset	$\|S\|$	$\|U\|$	$\|S \cup U\|$	$\|T\|$
MNIST	175	3325	3500	1500
H. Eggs	61	1176	1237	531
P. cysts	134	2562	2696	1156
H. Larvae (I)	122	2337	2459	1055
H. Eggs (I)	178	3400	3578	1534
P. cysts (I)	334	6363	6697	2871

propagation from S to U followed by supervised training on $S \cup U$ and testing on T — for statistical purposes.

After labels are propagated from S to U, we train a supervised classifier on $S \cup U$ using the latent feature space. For this task, we used the Optimum-Path Forest (OPF) [32] and Support Vector Machines (SVM) [33]. OPF has no hyperparameters to set, so it is simple to use. For SVM, we find optimal values for its hyperparameters σ (influence radius), C (regularization) and kernel type by grid search over the ranges [0.1, 0.000001], [1, 10000] and the kernel functions *Gaussian radial basis* and *linear* respectively, using 3 splits and stratified random sampling with 70% and 30% of the samples from $S \cup U$ used for training and validation, respectively.

We test the classifiers on T and measure their effectiveness by Cohen's κ coefficient [34]. The κ coefficient is within $[-1, 1]$, where $\kappa \leq 0$ means no agreement and $\kappa = 1$ means complete agreement between two annotators. Additionally, we also compute the accuracy of label propagation on U for each approach, that is the number of labeled samples correctly assigned divided by the number of unsupervised samples ($|U|$). Therefore, the best approach for label propagation is the one that produces the best supervised classifiers. Since we consider the κ as effectiveness measure, the best supervised classifier is then the one that provides the best κ result.

3.2. Baselines

As described in Section 2, we propose a semi-automatic label propagation (SALP) that uses OPF-Semi in the 2D *t*-SNE projection space to propagate labels to high-confidence samples and the user to propagate labels to low-confidence samples, respectively. We next compare SALP with the following three baselines:

1. *No label propagation (NLP):* SVM and OPF, are trained from only *S*, ignoring set *U*.
2. *Automatic label propagation (ALP):* set *U* is fully labeled by one of the four ALP methods below and SVM and OPF are trained from $S \cup U$.
   (a) LapSVM using the *n*D latent feature space.
   (b) LapSVM using the 2D *t*-SNE projection space.
   (c) OPF-Semi using the *n*D latent feature space.
   (d) OPF-Semi using the 2D *t*-SNE projection space.
3. *Interactive label propagation (ILP):* set *U* is fully labeled by the user and SVM and OPF are trained from $S \cup U$, as in [17].

In all above cases, we test SVM and OPF on *T*.

Table 2
Average κ and its standard deviation for SVM and OPF classifiers on the *T* set for MNIST, Parasites without impurity, and Parasites with impurity (I). The best results for each dataset are in bold.

Dataset	Propagation Technique	$\|S\|$	$\|U\|$	Average Propagation Accuracy	$\|S \cup U\|$	Average κ (SVM)	Average κ (OPF)
MNIST	No label prop.	175	-	-	175	**0.813415 ± 0.001**	0.709450 ± 0.021
	LapSVM (*n*D)	175	3325	0.095639	3500	0.000000 ± 0.000	0.051110 ± 0.006
	OPF-Semi (*n*D)	175	3325	0.763308	3500	0.736685 ± 0.053	0.716913 ± 0.048
	LapSVM (2D)	175	3325	0.574236	3500	0.521970 ± 0.065	0.580445 ± 0.047
	OPF-Semi (2D)	175	3325	**0.796592**	3500	0.780197 ± 0.008	**0.751794 ± 0.010**
H. Eggs	No label prop.	61	-	-	61	0.961366 ± 0.023	0.941358 ± 0.026
	LapSVM (*n*D)	61	1176	0.886338	1236	0.873472 ± 0.035	0.877344 ± 0.037
	OPF-Semi (*n*D)	61	1176	0.947563	1236	0.938317 ± 0.049	0.938323 ± 0.051
	LapSVM (2D)	61	1176	0.768141	1236	0.722691 ± 0.041	0.727673 ± 0.043
	OPF-Semi (2D)	61	1176	**0.982993**	1236	**0.983621 ± 0.011**	**0.982146 ± 0.008**
P. cysts	No label prop.	134	-	-	134	**0.823106 ± 0.016**	0.762682 ± 0.008
	LapSVM (*n*D)	134	2562	0.521598	2696	0.346761 ± 0.001	0.371770 ± 0.005
	OPF-Semi (*n*D)	134	2562	0.802238	2696	0.740287 ± 0.036	0.724182 ± 0.053
	LapSVM (2D)	134	2562	0.787666	2696	0.597541 ± 0.212	0.592956 ± 0.187
	OPF-Semi (2D)	134	2562	**0.838017**	2696	0.801953 ± 0.021	**0.786383 ± 0.022**
H. Larvae (I)	No label prop.	122	-	-	122	0.375378 ± 0.333	0.531080 ± 0.035
	LapSVM (*n*D)	122	2337	0.882613	2459	0.121253 ± 0.086	0.173416 ± 0.088
	OPF-Semi (*n*D)	122	2337	0.919075	2459	**0.601003 ± 0.073**	**0.602001 ± 0.066**
	LapSVM (2D)	122	2337	0.127086	2459	0.000000 ± 0.000	0.008665 ± 0.005
	OPF-Semi (2D)	122	2337	**0.924405**	2459	**0.569164 ± 0.070**	**0.556642 ± 0.059**
H. Eggs (I)	No label prop.	178	-	-	178	**0.705972 ± 0.037**	**0.568304 ± 0.034**
	LapSVM (*n*D)	178	3400	0.654118	3578	0.000000 ± 0.000	0.076043 ± 0.016
	OPF-Semi (*n*D)	178	3400	0.504510	3578	0.373763 ± 0.029	0.391894 ± 0.021
	LapSVM (2D)	178	3400	0.146274	3578	0.086608 ± 0.031	0.109734 ± 0.035
	OPF-Semi (2D)	178	3400	**0.729608**	3578	0.611144 ± 0.071	**0.544552 ± 0.047**
P. cysts (I)	No label prop.	334	-	-	334	**0.628584 ± 0.024**	**0.476051 ± 0.010**
	LapSVM (*n*D)	334	6363	**0.622662**	6697	0.232800 ± 0.104	0.202826 ± 0.030
	OPF-Semi (*n*D)	334	6363	0.468804	6697	0.343356 ± 0.034	0.337168 ± 0.032
	LapSVM (2D)	334	6363	0.128818	6697	0.045538 ± 0.038	0.079854 ± 0.024
	OPF-Semi (2D)	334	6363	0.605427	6697	0.429731 ± 0.013	0.396645 ± 0.009

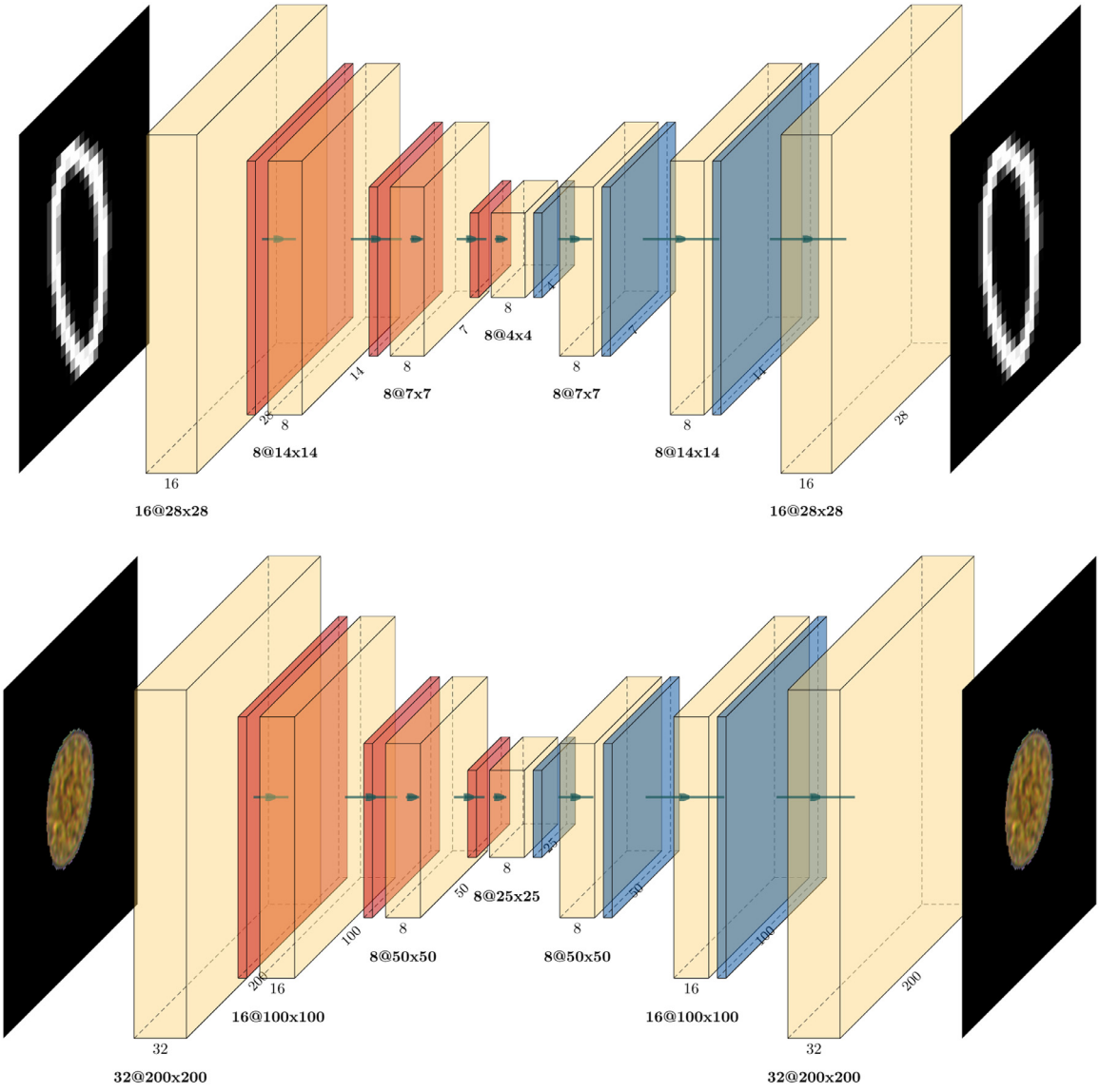

Fig. 5. AuNN architecture for MNIST dataset (top) and for Parasites datasets (bottom). The yellow layers are the convolutional layers, the red layers at the beginning of each network are the Max Pooling layers and the blue layers are the Up-Sampling layers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.3. Datasets

Our first dataset contains 5000 images (28 × 28 pixels each) of handwritten digits from 0 to 9, randomly selected from the popular public dataset MNIST [35]. Our next three datasets use images (200 × 200 pixels each) from an automatic processing pipeline that separates microscopy images of human intestinal parasites into three groups: (i) *Helminth larvae* and fecal impurities (3514 images); (ii) *Helminth eggs* and fecal impurities (5112 images); and (iii) *Protozoan cysts* and fecal impurities (9568 images).Fecal impurity is a diverse class that has very similar samples to parasites (see Fig. 4). We consider these three datasets with and without images of fecal impurities, yielding five datasets for testing our proposal, apart from MNIST. The number of classes in each dataset is as follows: (i) *H. Larvae* has two categories; (ii) *H. Eggs* has nine categories (*H. nana, H. diminuta, Ancilostomideo, E. vermicularis, A. lumbricoides, T. trichiura, S. mansoni, Taenia*, and impurities); and (iii) *P. cysts* has seven categories (*E. coli, E. histolytica, E. nana, Giardia, I. butschlii, B. hominis*, and impurities), respectively. Those are the most common species of human intestinal parasites in Brazil, which are responsible for public health problems in most tropical countries [36]. All three datasets are unbalanced with considerably more impurity samples. The images of parasites have been annotated by biomedical specialists. Table 1 gives the number of images in each set *S, U*, and *T* after the random split described in Section 3.1.

3.4. Implementation details

Feature extraction: Fig. 5 shows the AuNN architectures for the MNIST and parasites datasets. We implemented these networks in Keras [37] with 6 convolutional layers of 3 × 3 filters, 3 for the encoder and 3 for the decoder, respectively. After each convolutional layer, we use *ReLU* activation and apply max-pooling in the encoder and upsampling in the decoder. We normalize the input images within [0,1], since the output requires sigmoid activation. We choose the number of filters based on the dataset: For MNIST, the 6 convolutional layers use 16, 8, 8, 8, 8, and 16 filters. For the 5 parasites datasets, we use 32, 16, 8, 8, 16, and 32 filters respectively. As cost function, we use mean squared error as it provides more suit-

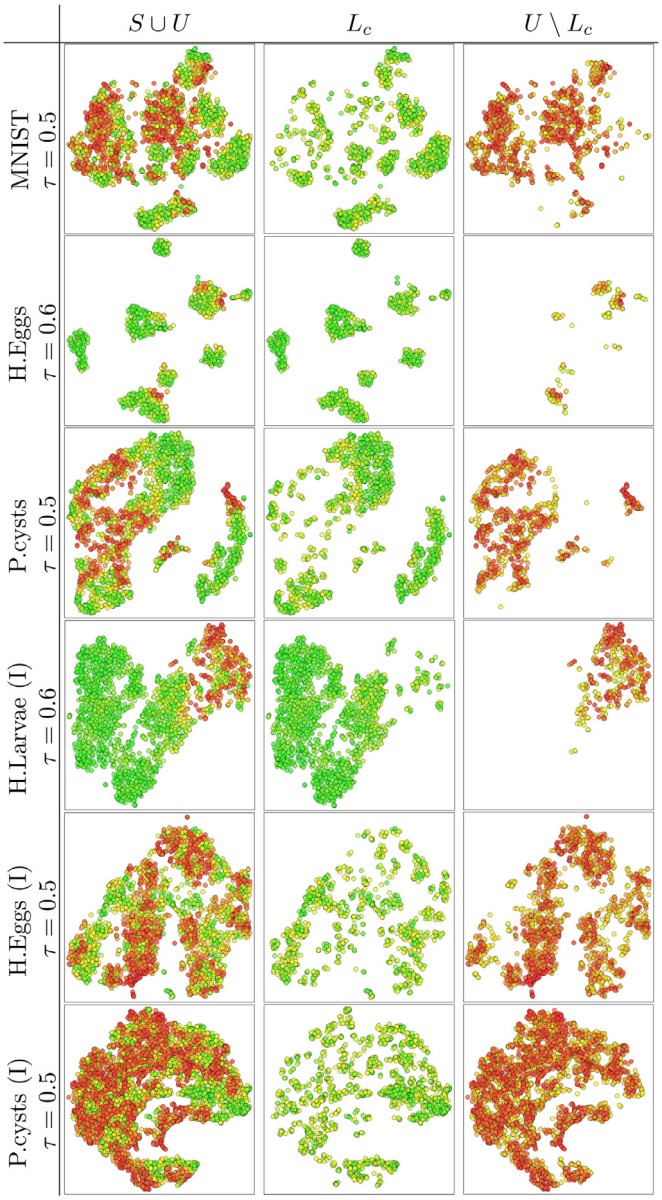

Fig. 6. Projections colored by label confidence (red=low confident, green=high confident). Rows are datasets (easiest at top, hardest at bottom). Columns show the entire set of supervised-and-unsupervised samples $S \cup U$, the high-confidence samples L_c labeled by ALP, and the low-confidence samples $U \setminus L_c$ that go to manual labeling. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

able results in reconstruction task with fewer training epochs. We use 50 epochs for the easier datasets (MNIST and H. Eggs without impurities) and 100 for the others. For MNIST, we use a latent feature space of $n = 128$ dimensions. For the parasites, which have higher-resolution and more complex images, we use $n = 5000$ dimensions. **Projection:** Different choices of t-SNE parameters can lead to different 2D projections [38]. We found empirically that for a range of 1000 to 7000 samples in $S \cup U$, setting t-SNE's perplexity to 40 and maximum iteration count to 1000 respectively yields good projections for label propagation.

3.5. Experimental results

We discuss the performance of our pipeline, measured by the performance of the classifiers trained from $S \cup U$ in the latent feature space and tested on T, by answering the following questions:

- Which space (nD latent, 2D projection) is better for ALP? (Section 3.5.1)
- How to set the confidence threshold τ? (Section 3.5.2)
- Which approach (manual, semi-automatic, automatic) best propagates labels from S to U? (Section 3.5.3)
- What is the end-to-end value of SALP? (Section 3.5.4)
- How do results depend on the projection quality? (Section 3.5.5)

Note that we use the 2D projection space only for manual label propagation, i.e. not for testing, since we cannot assume that set T is known during training.

3.5.1. Influence of reducing the feature space from n*D* to 2*D*

Table 2 presents mean and standard deviation values of Cohen's κ for classifiers on set T for each dataset, as well as the sizes of S, U, and $S \cup U$, and the mean accuracy values in automatic label propagation for LapSVM and OPF-Semi, used in the nD feature space and also in the 2D projection space, as well as the option of not propagating labels. We get several insights. First, we see that LapSVM performs sometimes better and sometimes worse in nD as compared to 2D, depending on the dataset. In contrast, OPF-Semi consistently shows a positive impact of reducing the feature space independently of the dataset. This happens even when its label-propagation performance is not the best one.

3.5.2. The choice of the confidence threshold

As stated in Section 2.3, users need to choose the threshold τ to specify which automatically-propagated labels they want to keep and which they wish to 'override' manually. Fig. 6 show the projections of all, respectively the most-confident samples selected by the user, for the six studied datasets. We see that the threshold τ varies relatively little (being either 0.5 or 0.6) across datasets. This indicates that a good default value to start with is $\tau = 0.5$, after which users can tune τ upwards or downwards depending on the actual distribution of confidences in the projection. Overall, we can see that the more challenging is the dataset, the higher is the threshold τ.

3.5.3. Best label propagation approach

Table 2 showed that OPF-Semi 2D is the winner for automatic label propagation (ALP). Hence, the next question is how well this method would compare against interactive label propagation (ILP) [17], which uses manual label propagation to *all* unsupervised labels, and our new semi-automatic label propagation (SALP), which uses manual label propagation to samples with low-confidence unsupervised labels only. Fig. 7 illustrates the ILP and SALP projections for the studied datasets. A key advantage of SALP over ILP is that it shows only the least confident samples (according to OPF-Semi 2D) to the user, hence reducing the effort needed to understand the picture (and also reducing clutter and overlap in the projection), thus making the interactive labeling task easier. We discuss next several observations relating ILP to SALP in Fig. 7, as well as observations we made during the actual interactive labeling process.

For the MNIST dataset, the user propagated labels to 1864 unsupervised samples on average (over the three considered runs) when using ILP. When using SALP, this number dropped to 1182 samples. This pattern of less effort for SALP is consistent over all other datasets, as discussed next.

For the *H. Eggs* dataset, we see that the ILP projection shows well-separated sample groups from distinct classes (colors). This indicates that separating classes in feature space is relatively easy. This is confirmed in turn by the fact that we only have very few low-confidence samples after running OPF-Semi 2D (black dots in the SALP projection). Hence, while labeling in ILP can proceed very

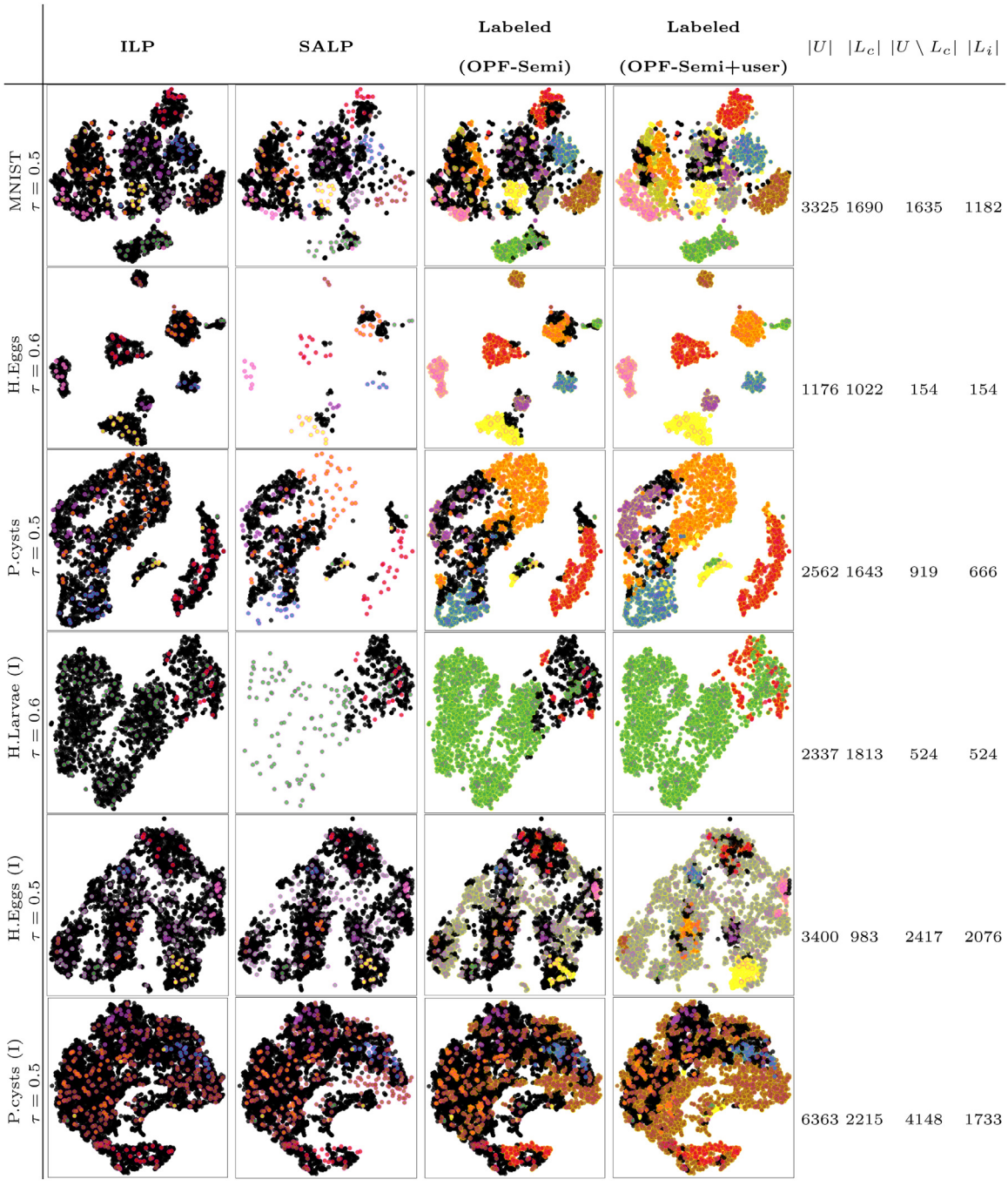

Fig. 7. Comparison of different label propagation methods (columns) for different datasets (rows). From left to right: ILP, SALP, labels automatically propagated by OPF-Semi, and final labeling result of SALP together with OPF-Semi. Colors indicate labels given by either supervised samples (ILP, SALP) or both unsupervised and propagated labels (OPF-Semi, OPF-semi+user)). Black shows samples to be considered by manual propagation (three left columns), and samples skipped by manual propagation (right column). Sample set sizes are shown to the right.

easily, given the good cluster separation, labeling in SALP is *even easier*, since we have both good cluster separation *and* a low number of samples to label. In this case, the user propagated labels to 1171 samples in ILP and to only 154 samples in SALP.

For *P. cysts*, the projections a less clear visual separation of same-class (same color) points in groups. This makes interactive label propagation more challenging for both ILP and SALP. The user propagated labels to 1999 samples in ILP and to 919 samples in SALP. For SALP, we see that OPF-Semi 2D propagated labels in more central regions of the visible groups where, hence, confidence is high. The remaining confusion regions (black points) are solved by the user.

For. *H. Larvae*, we notice that supervised impurity samples (green) are all over the projection, whereas the supervised *H. Larvae* samples (red) are more concentrated in the top-right of the projection. Given this quite good visual separation, propagating the impurity label using ILP is relatively easy for most parts of the projection. However, this still takes manual effort. Using SALP, such 'easy' areas are solved automatically, and the user is left with only the more difficult region at the top-right, where green meets red, to solve. In ILP, the user propagated labels for 2080 samples on average while in SALP this number was 524 samples.

For *H. Eggs* dataset with impurity, the supervised impurity samples (gray) fall between groups of colored points (actual *H. Eggs*

classes) in the projection. In contrast to the earlier datasets, we see many more black points in SALP, meaning that OPF-Semi 2D has difficulties in automatically propagating labels. This matches the fact that datasets with impurities are considerably harder. For this dataset, the user propagated labels to more points in SALP (2076) than ILP (1787). This seems to support the evidence that the simplification of the SALP projection by removing high-confidence points, even though minor in this case, was enough to help the user see more structure in the projection along which she could propagate labels. Also, as for *P. cysts*, we see that OPF-Semi 2D propagates labels in more central regions of the visible groups, leaving the rest to the user.

Finally, for *P. cysts* with impurities, the supervised impurity samples (brown) are spread out over the entire projection. The supervised *P. cysts* samples (other colors than brown) are mixed quite strongly, and the projection shows little structure – roughly, one large and one small crescent-shaped group. This is the most challenging dataset for manual label propagation and classification among the evaluated datasets. This difficulty can be noted by comparing *P. cysts* and *H. Eggs* both without impurities. For *P. cysts*, even without impurities, the classes are mixed in the projection. However, the classes are well separated in the projection for the *H. Eggs* dataset without impurities. When adding the impurities to those datasets, the difficulty increases for the classifiers, as shown in Section 3.5.4.

As for *H. Eggs*, OPF-Semi 2D finds only few confident samples, so the manual labeling effort is quite similar for both ILP and SALP. This is matched by the actual number of points to which the user actually propagated labels (1787 with ILP *vs* 1733 with SALP). Even though these figures are almost identical, the main benefit for SALP here is that OPF-Semi 2D already filtered the easy cases (high confidence) points, thereby focusing the user's effort to the more difficult cases.

3.5.4. End-to-end value of SALP

We have seen that SALP decreases the user's effort in label propagation. A final question we answer is: How much added-value does SALP bring, in terms of classification quality, as opposed to the earlier similar method, ILP, or to the best fully-automatic counterpart we found, OPF-Semi 2D? Table 3 answers this by showing the average and stardard deviation of κ on the test set T for each considered dataset. The table further shows the sizes of S, U, and $S \cup U$, and the mean accuracy values in label propagation for OPF-Semi 2D, ILP, and SALP. It is important to highlight that the propagation accuracy for SALP considers not only the low-confident samples labeled by the user, but the high-confident ones automatically treated by OPF-Semi 2D. We see that SALP consistently obtained the best classification results on unseen T for all datasets. This proves that SALP is, indeed, of added value with respect to earlier existing methods – using it yields better classifiers in the end. Separately, we see that, for all but the simplest datasets (MNIST and *H. Eggs*), SALP also yields the best label propagation accuracy.

3.5.5. How do results depend on projection quality

We did the same experiments discussed in the sections so far using UMAP [39] instead of *t*-SNE as a projection technique. Overall, we noticed worse results, in terms of label propagation accuracy and classifier quality (κ) than when using *t*-SNE. This indicates that the neighborhood preservation quality of a projection (which is higher for *t*-SNE than for UMAP) is am important factor for out method. Note also that the trends observed so far linking obtained SALP and ILP quality with the dataset size and difficulty cannot be ascribed to us having used 'optimal' projections by a lucky setting of the projection-method parameters: Indeed, both UMAP and *t*-SNE are non-deterministic methods.

4. Discussion

We next discuss several aspects of our method

4.1. Using the nD vs 2D feature space

An interesting question is how the fully automatic label propagation (ALP) performs when using the latent *n*D feature space *vs* the 2D projection space. Fig. 8 shows the average κ classification values for LapSVM and OPF-Semi using these two spaces for the OPF and SVM classifiers respectively. Datasets are sorted along the x axis by decreasing order of the κ value for OPF-Semi 2D. We see that LapSVM leads to better results in 2D than in *n*D for half of the datasets, while OPF-Semi does that for *all* datasets. This essentially tells that the 2D projection space, created by *t*-SNE, is able to retain all needed information to enable the desired label propagation and, next, good-quality classifier construction. This is an important result, as it justifies next presenting the 2D projection space to the user as the sole information based on which she will perform the manual label propagation. We also see that the trend of the κ values along the x axis, for both the 2D and *n*D variants, matches the perceived difficulty of the datasets: High κ values correspond to easier datasets (left), while lower κ values correspond to the harder datasets with impurities (to the right). Finally, we

Table 3
Average κ and its standard deviation for the SVM and OPF classification results on unseen test data T for the studied datasets. Best results per dataset are in bold.

Dataset	Propagation Technique	$\|S\|$	Average $\|L_i \cup L_c\|$	Average Propagation Accuracy	Average $\|S \cup L_i \cup L_c\|$	Average κ (SVM)	Average κ (OPF)
MNIST	OPF-Semi (2D)	175	3325	0.796592	3500	0.780197 ± 0.008	0.751794 ± 0.010
	ILP	175	1864	**0.974718**	2039	0.844264 ± 0.027	0.776241 ± 0.036
	SALP	175	2872	0.947192	3047	**0.885855 ± 0.030**	**0.839161 ± 0.018**
H. Eggs	OPF-Semi (2D)	61	1175	0.982993	1236	0.983621 ± 0.011	0.982146 ± 0.008
	ILP	61	1171	**0.996014**	1232	**0.986624 ± 0.009**	**0.987364 ± 0.003**
	SALP	61	1175	0.992347	1236	**0.989582 ± 0.005**	**0.983639 ± 0.007**
P. cysts	OPF-Semi (2D)	134	2562	0.838017	2696	0.801953 ± 0.021	0.786383 ± 0.022
	ILP	134	1999	0.947177	2133	0.851948 ± 0.006	0.841023 ± 0.002
	SALP	134	2309	**0.951119**	2443	**0.877566 ± 0.011**	**0.850232 ± 0.015**
H. Larvae (I)	OPF-Semi (2D)	122	2337	0.924405	2459	0.569164 ± 0.070	0.556642 ± 0.059
	ILP	122	2080	0.981273	2202	0.727843 ± 0.013	0.723049 ± 0.016
	SALP	122	2337	**0.986730**	2459	**0.805388 ± 0.014**	**0.748340 ± 0.036**
H. Eggs (I)	OPF-Semi (2D)	178	3400	0.729608	3578	0.611144 ± 0.071	0.544552 ± 0.047
	ILP	178	1547	0.914358	1725	0.683544 ± 0.033	0.593104 ± 0.034
	SALP	178	3059	**0.959611**	3237	**0.866121 ± 0.043**	**0.725803 ± 0.025**
P. cysts (I)	OPF-Semi (2D)	334	6363	0.605427	6697	0.429731 ± 0.013	0.396645 ± 0.009
	ILP	334	1787	0.826867	2121	0.589643 ± 0.036	0.472148 ± 0.008
	SALP	334	3948	**0.864390**	4282	**0.648831 ± 0.043**	**0.543963 ± 0.016**

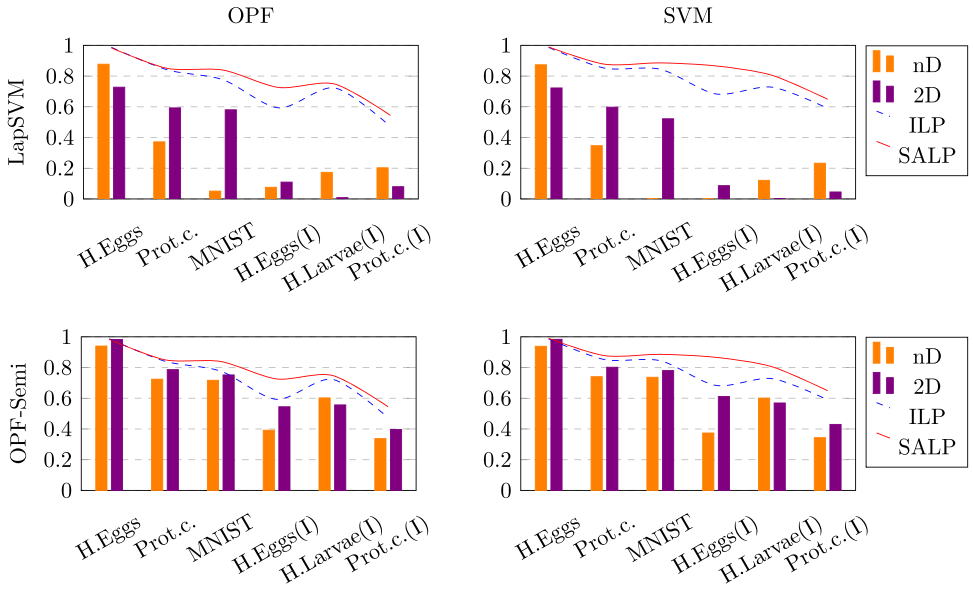

Fig. 8. κ values for studied datasets, for OPF and SVM classifiers (columns) and LapSVM and OPF-Semi automatic label propagation methods (rows). Curves show κ values for ILP and SALP for comparison purposes. Datasets are sorted from easiest to hardest to classify (left to right) based on SALP results.

plot here also the κ values for ILP and SALP (curves in the figures). In all cases, these curves are above the automatic methods, showing that adding manual effort pays off. The SALP curve is above the ILP one, showing that the optimal design is reached by *combining* automatic and manual label propagation (both executed in the 2D space).

4.2. User effort reduction

Besides achieving the best classification results, as compared to both fully-automatic and fully-manual (ILP) label propagation, SALP also reduces the *manual* effort as compared to ILP. Fig. 9 shows this by depicting the percentage of samples labeled by the user over total number of samples to label ($|U|$) per dataset and for ILP and SALP. For SALP, this measurement excludes, indeed, the automatically-labeled samples by OPF-Semi 2D. Datasets are sorted along x by increasing $|U|$, i.e, from the smallest to the largest dataset. Fig. 9 reveals several insights. First, assuming that the labeling effort is proportional with the number of labeled samples and the effort per sample is the same for ILP and SALP (which should be the case given that the two methods share the same visualization and interaction), we see that the ILP effort is always larger than the SALP effort, except for *H. Eggs* with impurities. Secondly, the percentage of propagated samples for ILP decreases with the dataset size. This can be explained by the difficulty of propagating labels in projections showing many points, where overlap and clutter become issues. We note an opposite for trend SALP: The percentage of propagated samples increases with dataset size. The trend breaks for the largest dataset (Prot.c.(I), 6363 samples), about twice larger than the second-largest dataset (*H. Eggs*(I), 3400 samples). Here, the projection is likely quite dense and cluttered, so manual propagation becomes similarly hard for ILP and SALP.

In parallel, we observe that the number of samples $U\backslash L_c$, those above the threshold τ and low-confidence labels to OPF-Semi, also increases with the dataset size. Thus, the amount of samples $U\backslash L_c$ presented to the user to propagate labels with SALP increases with dataset size. One case in point is the *H. Eggs* with impurities dataset. This dataset has the largest percentage of annotated

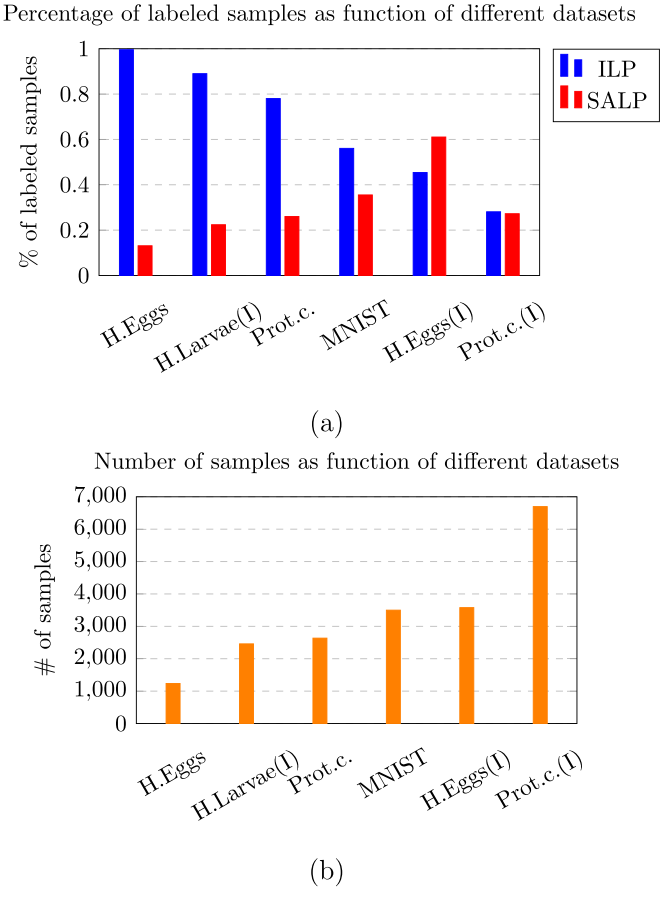

Fig. 9. (a) the percentage of labeled samples in U *vs* ILP and SALP label propagation methods and (b) the number of samples for the six studied datasets.

samples by SALP, exceeding also ILP. This is explained by the size of the dataset (second largest one) and the fact that its projection makes it reasonably easy to propagate labels for the large impurity class (Fig. 6).

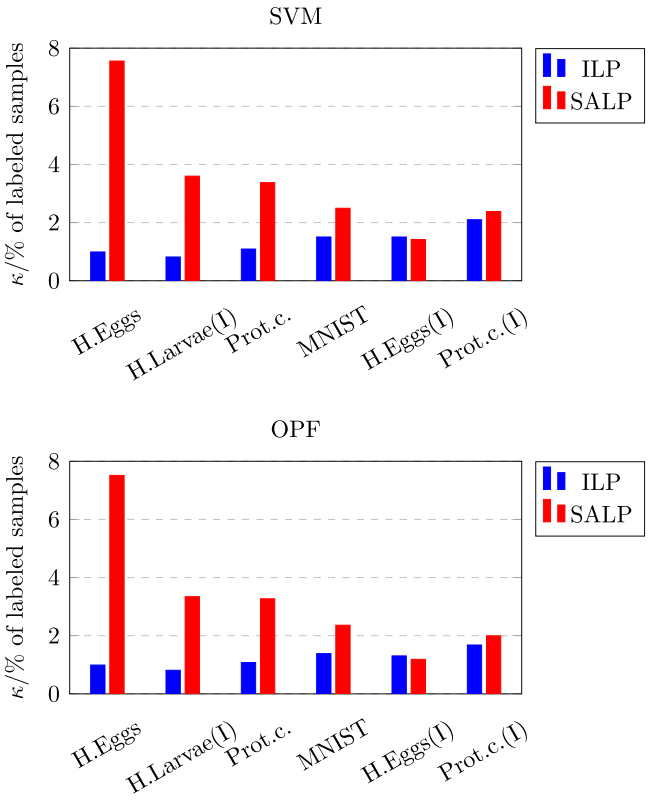

Fig. 10. Normalized gain, i.e., κ divided by the percentage of manually labeled samples with ILP and SALP for SVM and OPF classifiers for the six studied datasets (sorted left to right on size).

4.3. Effectiveness

As shown in Fig. 8, SALP consistently yields best classification results, for both SVM and OPF classifiers, overpassing fully manual propagation methods (ILP) and the best fully automatic one (OPF-Semi 2D). The gains of SALP are higher for the more challenging datasets, where fully automatic methods encounter challenges. Conversely, where such methods work well, they reduce user effort as compared to fully manual propagation (ILP). In brief, this shows that the combination of automatic methods with human insights is indeed of added value both in increasing classifier quality and decreasing the effort needed to achieve it.

It is next interesting to compare the *normalized gain* of ILP *vs* SALP. We define this as the obtained κ value (what we get) divided by the percentage of manually labeled samples (what we need to pay). Fig. 10 shows this normalized gain for ILP and SALP for both SVM and OPF classifiers. We see that SALP has far larger normalized gains than SALP for smaller datasets, while differences become quite small for the two largest datasets.

4.4. Manual sample selection justification

In classical pipelines, expert users would label samples in an empirical order. In pipelines that consider active learning methods, the sample informativeness can be used to suggest samples in each iteration for user supervision. However, those approaches do not usually explore the ability of humans in abstracting information from data visualization. Given that their labeling effort is limited (and their cost is high), the aim is to maximize the 'added value' of creating extra labels manually. Our hypothesis (which we show, by our experiments, to hold) is that, when expert users are offered hints in terms of sample similarity (via the 2D projection and its tooltips) and by the confidence of an automatic labeler (color-coded in the projection), they can manually create extra labels that have a higher added-value (for classification accuracy) than fully automatic methods can achieve.

The core point of manual labeling is to enable users with expert knowledge select the samples they think are most relevant for constructing a good training set. Answering the question of why expert users would select a certain sample subset rather than another one is not something we can argue theoretically, as it depends on a multitude of factors – first and foremost, the training of the expert and how this training determines the expert to consider a given image more (or less) relevant for being labeled in a certain way.

4.5. Limitations

Several limitations exist to our approach, as follows. First, *validation* is limited to six datasets, two classifier techniques, and one user performing manual labeling. Measuring the added-value of SALP for more (dataset, classifier, user) combinations would bring more insights into the effectiveness of the method. Secondly, while the added-value of the 2D *t*-SNE projection space in capturing information needed for good label propagation has been demonstrated both for automatic methods and manual ones, the actual effect of *t*-SNE's distortions has not been quantitatively gauged. Using projection accuracy metrics such as stress, trustworthiness, continuity, or neighborhood hit [31] can be used to find such correlations. On the other hand, using visual tools [31] that highlight such errors in specific projection areas can help the user to achieve more accurate and/or faster manual label propagation.

5. Conclusion

We proposed a combined automatic-and-user-driven approach for creating labeled samples for sparsely-annotated datasets for the purpose of training classifier models. For this, we extract dataset features using Autoencoder Neural Networks and next reduce these to a 2D space using *t*-SNE. We next automatically propagate labels from the (few) supervised to unsupervised samples in this 2D space, while monitoring the propagation confidence. For low confidence labeled samples, we allow the user to manually annotate them by using the visual insights encoded in the 2D projection annotated with the supervised sample labels. Several quantitative results follow: First, we showed that the 2D projection space leads to higher-accuracy automatic label propagation than the high-dimensional latent space extracted by the autoencoder. To our knowledge, this insight is new, and suggests new ways for dimensionality reduction. Secondly, we show that our semi-supervised method, combining the OPF-Semi automatic label propagation with user-driven manual label propagation, both done in the 2D space, achieves higher classification quality than both fully-automatic and fully-manual label propagation. This opens the way to different methods for combining automatic and human-centered methods for the engineering of high-quality machine learning systems.

Future work will consider the use of the proposed semi-automatic label propagation method in Active Learning (AL) scenarios. We expect that AL looping can improve classification results as long as the propagation accuracy increases. Also, we intend to consider metric learning approaches that might improve the 2D projection of the feature space. We are interested in methods that allow the comparison between training and testing data. Specifically, we intend to investigate methods such as the exemplar-centered High Order Parametric Embedding [40]. Separately, we plan to perform more extensive validation studies measuring the

added-value of our approach for more types of datasets, classification methods, and using additional visual analytics techniques to help users to propagate labels better and faster.

Acknowledgments

The authors are grateful to FAPESP grants #2014/12236-1, #2016/25776-0 and #2017/25327-3, and CNPq grants 303808/2018-7. The views expressed are those of the authors and do not reflect the official policy or position of the São Paulo Research Foundation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Krizhevsky, I. Sutskever, H.E. Geoffrey, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.
[2] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[3] S. Levine, C. Finn, T. Darrell, P. Abbeel, End-to-end training of deep visuomotor policies, J. Mach. Learn. Res. 17 (1) (2016) 1334–1373.
[4] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[5] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[6] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, in: NIPS'14, MIT Press, Cambridge, MA, USA, 2014, pp. 3320–3328.
[7] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[8] R. Mash, B. Borghetti, J. Pecarina, Improved aircraft recognition for aerial refueling through data augmentation in convolutional neural networks, in: Proc. ISVC, Springer, 2016, pp. 113–122.
[9] S.J. Nowlan, G.E. Hinton, Simplifying neural networks by soft weight-sharing, Neural Comput. 4 (4) (1992) 473–493.
[10] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 3581–3589.
[11] G. Forestier, C. Wemmert, Semi-supervised learning using multiple clusterings with limited labeled data, Inf. Sci. 361–362 (2016) 48–65.
[12] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, K. Talwar, Semi-supervised knowledge transfer for deep learning from private training data, in: Proceedings of the International Conference on Learning Representations, 2017.
[13] P. Rauber, A. Falcão, A. Telea, Projections as visual aids for classification system design, Inf. Vis. (2017).
[14] P.E. Rauber, A.X. Falcão, A.C. Telea, Visualizing time-dependent data using dynamic t-SNE, in: Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Short Papers, in: EuroVis '16, 2016, pp. 73–77.
[15] A.Z. Peixinho, B.C. Benato, L.G. Nonato, A.X. Falcão, Delaunay triangulation data augmentation guided by visual analytics for deep learning, in: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018, pp. 384–391.
[16] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, M. Sedlmair, Comparing visual-interactive labeling with active learning: an experimental study, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 298–308.
[17] B.C. Benato, A.C. Telea, A.X. Falcão, Semi-supervised learning with interactive label propagation guided by feature space projections, in: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018, pp. 392–399.
[18] B. Settles, Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
[19] S. Patra, L. Bruzzone, A batch-mode active learning technique based on multiple uncertainty for SVM classifier, IEEE Geosci. Remote Sens. Lett. 9 (3) (2012) 497–501.
[20] P.A.V. Miranda, A.X. Falcão, Links between image segmentation based on optimum-path forest and minimum cut in graph, J. Math. Imaging Vis. 35 (2) (2009) 128–142.
[21] T.V. Spina, P.A.V. Miranda, A.X. Falcão, Intelligent understanding of user interaction in image segmentation, Int. J. Pattern Recognit. Artif. Intell. 26 (02) (2012) 1265001.
[22] A.T. Silva, J.A. Santos, A.X. Falcão, R.S. Torres, L.P. Magalhães, Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning, Comput. Vis. Image Underst. 116 (4) (2012) 510–523.
[23] L.V.D. Maaten, Accelerating t-SNE using tree-based algorithms, J. Mach. Learn. Res. 15 (1) (2014) 3221–3245.
[24] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: Proc. Intl. Conf. on Artificial Neural Networks (ICANN), Springer, 2011, pp. 52–59.
[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[26] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[27] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[28] V. Sindhwani, P. Niyogi, M. Belkin, Beyond the point cloud: from transductive to semi-supervised learning, in: Proceedings of the 22Nd International Conference on Machine Learning, in: ICML '05, ACM, New York, NY, USA, 2005, pp. 824–831.
[29] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[30] W.P. Amorim, A.X. Falcão, J.P. Papa, M.H. Carvalho, Improving semi-supervised learning through optimum connectivity, Pattern Recognit. 60 (C) (2016) 72–85.
[31] L. Nonato, M. Aupetit, Multidimensional projection for visual analytics: linking techniques with distortions, tasks, and layout enrichment, IEEE TVCG (2018).
[32] J.P. Papa, A.X. Falcão, V.H.C. Albuquerque, J.M.R.S. Tavares, Efficient supervised optimum-path forest classification for large datasets, Pattern Recognit. 45 (1) (2012) 512–520.
[33] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (4) (1998) 18–28.
[34] J.L. Fleiss, J. Cohen, The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability, Educ. Psychol. Meas. 33 (3) (1973) 613–619.
[35] Y. LeCun, C. Cortes, MNIST handwritten digit database, 2010.
[36] C.T.N. Suzuki, J.F. Gomes, A.X. Falcão, S.H. Shimizu, J.P. Papa, Automated diagnosis of human intestinal parasites using optical microscopy images, in: 2013 IEEE 10th International Symposium on Biomedical Imaging, 2013, pp. 460–463.
[37] F. Chollet, et al., Keras, 2015, (https://keras.io).
[38] M. Wattenberg, F. Viegasand, I. Johnson, How to use t-SNE effectively, Distill (2016).
[39] L. McInnes, J. Healy, J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, ArXiv e-prints (2018).
[40] M.R. Min, H. Guo, D. Song, Exemplar-centered supervised shallow parametric data embedding, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, in: IJCAI17, AAAI Press, 2017, p. 24792485.

Bárbara C. Benato received her MSc (2019) in computer science from University of Campinas, Brazil. She is currently PhD student in Computer Science at University of Campinas, Brazil. Her research interests include machine learning, deep learning, pattern recognition and visual analytics applications in machine learning.

Jancarlo F. Gomes received his PhD (2008) in Parasitology from the University of Campinas, Brazil. He is currently Professor at School of Medical Sciences, University of Campinas. His research interests include public health, human and veterinary parasitology, parasites diagnosis (conventional/automated) and image processing.

Alexandru C. Telea received his PhD (2000) in computer science from the Eindhoven University of Technology, the Netherlands. He is currently a professor in visual data analytics at the Department of Information and Computing Sciences, Utrecht University. His interests include 3D multiscale shape processing, information visualization, software visualization, machine learning, and visual analytics.

Alexandre X. Falcão received his PhD (1996) in electrical engineering from the University of Campinas, Brazil. He has been Professor at the Institute of Computing, University of Campinas and his research interests include image segmentation and analysis, volume visualization, content-based image retrieval, mathematical morphology, digital TV, medical imaging applications and pattern recognition.