cVIL: Class-Centric Visual Interactive Labeling

Matthias Matt[1] , Matthias Zeppelzauer[2] , Manuela Waldner[1]

[1]TU Wien, Institute of Visual Computing & Human-Centered Technology, Austria
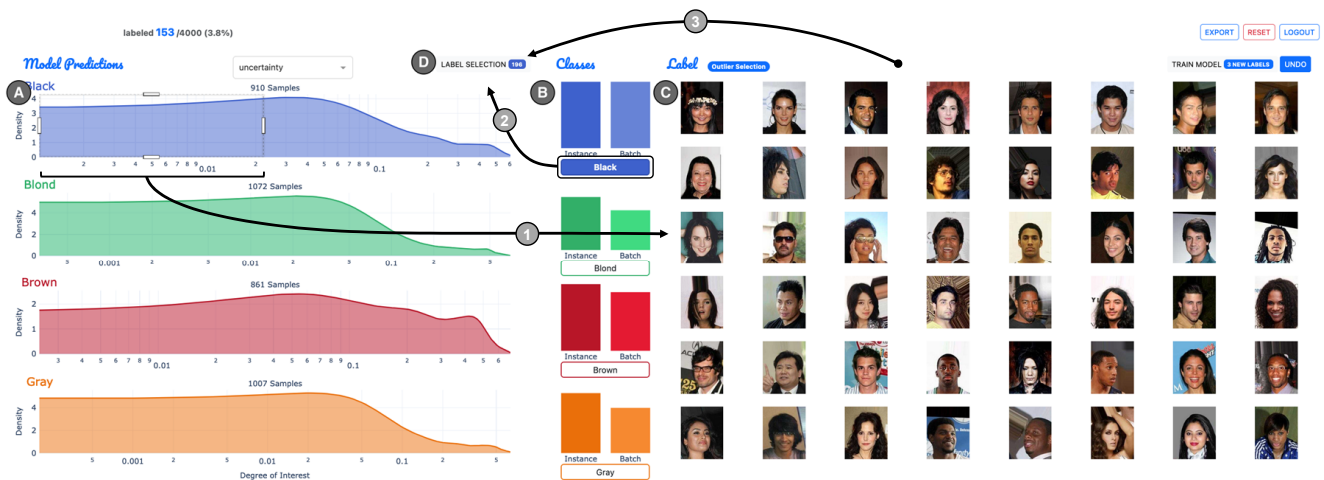[2]St. Pölten University of Applied Sciences, Austria

Figure 1: *Batch labeling workflow using cVIL for labeling 4,000 images according to hair color: (1) selecting a range of images with low uncertainty for black hair from the class density plot (A) yields a preview of images with the highest uncertainty in the selection (C). (2) The anticipated target class ("black") is selected from the class distribution view (B), which shows the proportion of instances already labeled individually (instance labels) or through a range selection (batch labels). (3) The user confirms the target class (if all persons have black hair) by labeling all images in (C) in a batch manner (D) – or updates the selection (if not all persons have black hair; A).*

Abstract
We present cVIL, a class-centric approach to visual interactive labeling, which facilitates human annotation of large and complex image data sets. cVIL uses different property measures to support instance labeling for labeling difficult instances and batch labeling to quickly label easy instances. Simulated experiments reveal that cVIL with batch labeling can outperform traditional labeling approaches based on active learning. In a user study, cVIL led to better accuracy and higher user preference compared to a traditional instance-based visual interactive labeling approach based on 2D scatterplots.

CCS Concepts
• Human-centered computing → Visual analytics; User interface design;

1. Introduction

Reducing costs for human annotation of unlabeled, unstructured data, such as image collections, has been widely studied. One standard methodology originating from the domain of machine learning is active learning (AL), where the model steers the labeling process [Set09]. One prominent AL strategy is uncertainty-based selection where the most ambiguous instance is selected for labeling [FZL12].

Using AL, the user's role is, however, limited to labeling those instances, that are autonomously selected by the system, which can easily get tedious and frustrating to users [ACKK14]. A successful approach for integrating user initiative in the labeling process is visual interactive labeling (VIL). VIL has been found superior to AL at least for simple labeling tasks [BHZ*18]. VIL systems typically show parts or all data in a visualization, where the user can interactively select instances for labeling. For example,

Seifert and Granitzer [SG10] show the model confidence for all instances and classes in a radial visualization. VIL has been extended by guidance mechanisms [GCT22] and semi-automatic labeling approaches based on clustering [CBB*19, BT20]. Some systems also allow interactively selecting a threshold based on confidence [BGTF21] to label samples or other predefined metrics to grow clusters based on distance [SKK20] or purify clusters based on outlier scores [BT20]. To combine the strengths of VIL and AL, Bernard et al. [BZSA18] have introduced a process model, which allows both the model and user to select instances for labeling.

Typically, VIL approaches, as described above, are *instance-centric*, i.e., they show the class assignment of all data instances to be labeled in some spatial representation like a 2D scatterplot. We refer to such systems as iVIL. When working with large amounts of data, it is impractical to interact with the data directly. [CTBA*11]. Consequently, the visualization in a scatterplot can limit the system's scalability with respect to the amount of data to be visualized and labeled – especially if the data distribution in the projection space is complex and classes strongly overlap in the projection [NA18]. Furthermore, the preceding dimensionality reduction may introduce (non-linear) distortions. Figure 2 illustrates the limitation of scatterplots: It shows t-SNE [vdMH08] projections of two subsets of the CelebA [LLWT15] dataset (which we call Celeb-Hair and CelebGlasses), where the classes refer to a person's hair color (upper plot) and whether or not they wear glasses (lower plot). Each dataset comprises 1,000 images per class. The images are represented by DINO embeddings [CTM*21]. Color encodes the ground truth class for illustration. It is easy to see that labeling this data (imagine all points would be gray because no labels exist yet) becomes a complex and tedious task. Selecting individual instances is time-consuming, and selecting multiple instances of one class at once is difficult, due to the large class overlaps.

To address the above challenges, we propose a *class-centric* VIL approach, named cVIL. The main contributions are:

- A new visual interactive labeling approach, which combines principles of AL (the computation of well-understandable properties) and VIL (leaving the initiative for selecting items to be labeled to the user; Section 2.1).
- A labeling interface that scales independently of the number of data instances (Section 2.2) and allows for efficient instance and batch labeling (Section 2.3).
- Results from simulations that demonstrate better efficiency compared to AL (Section 3) and from a user study that show higher labeling accuracy and user preference compared to iVIL (Section 4).

2. Class-Centric Visual Interactive Labeling

cVIL provides a visual overview of class-centric property measure values [BHS*21] from which users can select instances or batches of instances for labeling. By showing distributions of property measures across instances at a class level rather than features per instance, cVIL disentangles the visual representation from the number of instances and helps users to interactively train a classifier guided by property measures. This can speed up the labeling process considerably.
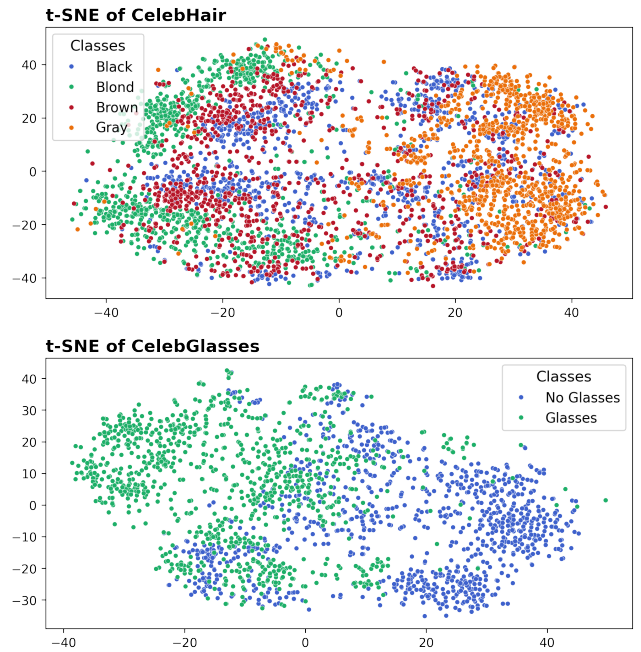

Figure 2: *t-SNE projection of high dimensional embeddings found by representation learning, used for the iVIL approach in our user study.*

2.1. Measures

By observing participants during an instance-centric visual labeling process, Bernard et al. [BHZ*18] found that participants use specific strategies while labeling. These strategies were later formalized into the concept of *property measures* [BHS*21]. For cVIL, we experimented with three representative and complimentary property measures, which either use the model output, data distribution, or both to calculate the final result:

Min-Margin: This property measure is based on class probability, defining Min-Margin as the difference between the largest and second-largest class probability value. To get a measure of uncertainty, the resulting value is subtracted from 1. For example, given a class assignment $(0.1, 0.1, 0.5, 0.3)$ for a single instance, the uncertainty of this instance based on Min-Margin is given by $1 - (0.5 - 0.3) = 0.8$.

Eccentricity: We define sample eccentricity as the Euclidean distance to the overall median of the data. Before calculating the distance, each dimension is first scaled by the inverse of its respective variance. The variance is computed over the entire input data. This measure is completely data-centric (i.e. model-independent) in contrast to the others and enables the identification of outliers in the data distribution.

Disagreement: The average Jensen-Shannon distance between a data point's class probability distribution and that of all its neighbors defines its disagreement in relation to its neighborhood. The neighborhood of a sample is defined by the *k*-nearest neighbors based on Euclidean distance. While the neighbors of a sample are selected from the data space, the resulting property measure value

is calculated based on the class probability distribution of the neighbors, making it a hybrid approach. A value of $k = 20$ was chosen for all evaluations. This measure highlights instances in particularly inhomogeneous regions wrt. class assignments in the feature space.

2.2. Visual Encoding

The property measures output one scalar value per unlabeled data instance, whereby each data instance is assigned to the predicted class based on the probability output of the model for this instance. The visual encoding, therefore, only needs to convey a univariate distribution per class. To ensure scalability with the number of data instances, we chose an area plot based on kernel density estimation (KDE) of the values, as shown in Figure 1 (A). The x-axis represents the values of the respective property measure (e.g., Min-Margin), and the y-axis is the estimated density of instances corresponding to this value. The x-axes of the plots are not shared because only the relative position within each class partition is relevant for labeling.

2.3. Interaction Design

The KDE class distribution plots allow users to preview images by hovering or selecting images within a certain property measure range (see Figure 1). In the preview panel, images with the largest property measure values in the selection will then be displayed, which tend to represent the most uncertain images within the selection – and, therefore, the most important ones for the learning and labeling process. The number of image previews is limited to the size of the panel. Hovering over the KDE class distribution plot updates the preview images so that images with equal or lower property measure values are shown. This way, the user can quickly validate model predictions, making cVIL also suitable for *Quality Assurance for Machine Learning* (QA4ML) [ZTdJ*21] tasks.

After making a selection in the KDE plot and selecting a target class, users have two options for how they can assign the label of the selected target class to the images: using *instance labeling*, users click on individual images from the preview to label them with the selected class. This is the most useful interaction after a cold start and when selecting images after making an outlier selection (i.e., selecting images with a high uncertainty score for a particular class) to identify and label particularly difficult instances.

The second option is *batch labeling*, which can be used to quickly confirm class predictions for instances that are rather certain. Low property measure values are a good indicator for such instances. If users think that all instances within the selected range do belong to the selected target class, they just press a button to label all instances from the current selection in the KDE plot at once, as illustrated in Figure 1. Note that this may lead to the labeling of more images than are currently visible in the preview panel and, therefore, may produce wrong labels for out-of-class examples. To prevent this, users can validate the selection range with the hover preview to make sure the images are correct. In our user study (Section 4), we did not observe that batch labeling led to a critical number of incorrect labels. After model retraining, which took an average of three seconds in our user study, all KDE plots are updated with the new property measure values. Due to the relatively

short re-training time, users can easily update the model while labeling.

3. Quantitative Experiments: cVIL vs. AL

Firstly, we compared the property measures in terms of their runtime. When benchmarking our implementation of the three property measures with an M1 MacBook Pro, we found that – for an embedding size of 1,024 dimensions and up to more than 16,000 samples – the property measures for all instances can be calculated in less than a second. In particular, the results for Disagreement took close to one second to calculate, while Eccentricity took less than 100ms and Min-Margin only around 1ms.

To assess the performance of the selected property measures and the potential of cVIL, we compared the theoretical performance of cVIL (accuracy per simulated labeling step) to AL. For evaluation we use MNIST [LCB10], STL10 [CNL11], as well as CelebHair and CelebGlasses. For MNIST, the raw pixels are used as training data. For the more complex datasets, we use DINO [CTM*21] for representation learning for the simulations as well as the user study. The backbone model of DINO is a vision transformer, which was pre-trained on ImageNet [DDS*09].

We simulate instance labeling by selecting samples with the highest property measure value in each class partition and assigning them their correct ground truth label. An equal number of samples are selected from each class to get a total of ten samples per iteration. For example, for STL10, one sample with the highest property measure value per class is selected. We simulate optimal batch labeling by selecting instances with the lowest property measure value up to the first incorrectly predicted instance for labeling. Therefore, the number of samples in a batch selection can vary, depending on the accuracy of the model and property measures.

The classifier is a multi-layer perceptron with two hidden layers consisting of 50 and 20 neurons. Batch labels are given a lower sample weight during training, which is realized by assigning them lower costs in the loss function before back-propagation. The number of batch labels that are used for each class during training is determined by the number of samples C_{min} in the class with the fewest instance labels. In the evaluations, $10 \cdot C_{min}$ instances were then randomly sampled from the batch labels of each class. The weights of samples labeled through batch labeling were set to 0.1.

Based on the evaluation results with MNIST and STL10 datasets, shown in Figure 3, we observed that Disagreement and Eccentricity are less effective for instance labeling. Min-Margin leads to faster model improvement and is en par with AL simulations that also use Min-Margin. This indicates that class partitioning with instance labeling is comparable in performance to AL. For STL10, Eccentricity shows slightly better results than Disagreement, while the opposite is true for MNIST. For CelebHair and CelebGlasses, Eccentricity has a considerably lower accuracy, while all other methods – including AL – perform similarly. To summarize, instance labeling using cVIL performs best with Min-Margin and yields comparable performance to Min-Margin-based AL.

For batch labeling, we observe that Disagreement and Eccentricity slightly outperform Min-Margin for MNIST and both strate-
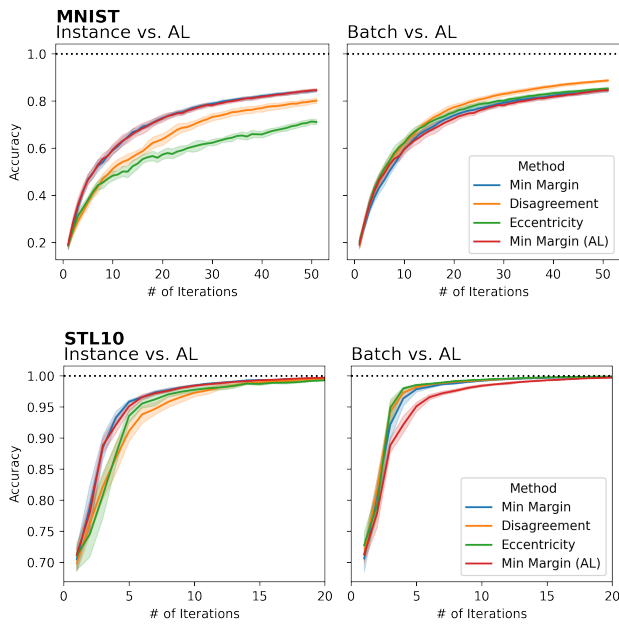
Figure 3: *cVIL with Min-Margin is en par with AL for instance labeling (left). For batch labeling, cVIL outperforms AL independently of the property measure (right).*

gies outperform AL. For CelebHair and CelebGlasses, all methods perform similarly well. For STL10, which is the most complex dataset, cVIL clearly outperforms AL with all three property measures, with peak performance for Eccentricity. To sum up, batch labeling can easily compete with AL and outperforms it, in particular for complex data, even when using the same measure (Min-Margin). Surprisingly, a purely data-driven strategy (Eccentricity) outperforms all others. Our experiments show that cVIL with batch labeling has clear potential to outperform traditional AL.

4. User Study: cVIL vs. iVIL

In the user study, we compare our class-centric VIL interface (cVIL) to a simple instance-centric VIL interface (iVIL). For the iVIL baseline interface, the KDE plots from Figure 1 (A) are replaced with a t-SNE [vdMH08] projection of the DINO feature vectors [CTM*21], as shown in Figure 2. The colors shown in the interface represent the *predicted* labels associated with the data instances. Users can either label individually selected instances or perform batch labeling using a lasso selection in the scatter plot. Our hypothesis is that cVIL is more effective at labeling than iVIL while being less demanding and more preferable to use.

4.1. Study Design

Participants: The user study was conducted with 16 participants (11 male, 4 female, 1 non-binary). All of them had a background in computer science, with 4 being post-graduate, 6 graduate, and 6 undergraduate level. Moreover, 11 of the participants had prior

experience with machine learning. All participants were between 22 and 40 years old with a median age of 26.

Task: During the study, the participants were given labeling tasks. The system was not initialized, which meant that participants had to first label samples based on a random selection to train the model at which point the full interface became available. The task was considered complete only when participants believed that all the data points had their correct label – either assigned manually by the user or predicted by the model. The participants had to decide when this stage was reached and were instructed to export the labels at that point. The entire study, including setup and introduction, was estimated to take approximately one hour to complete.

Data: We asked users to label the CelebGlasses and CelebHair data sets (see Figure 2). To reduce the complexity of the CelebHair dataset, a subset with only black and gray hair was used. To reduce the amount of time required for labeling, the number of samples was limited to 1,000 per class. For the user study, only Min-Margin was available to the participants due to its simplicity and good overall performance. This further reduced the task complexity by only presenting users with a single measure to focus on.

Independent and dependent variables: The study used a within-subjects design, with the interface (cVIL vs. iVIL) as the independent variable. We also varied the dataset assigned to the two interfaces, as well as the order of appearance of the interfaces. The overall accuracy of both manual and predicted labels, labeling time, cognitive demand, and preference were the dependent variables being observed.

Procedure: Participants were provided with a tutorial sheet that explained the system's components and how to interact with them prior to attempting the task. They solved the tasks on an external monitor and had access to a mouse and keyboard. The study was conducted on an M1 MacBook Pro. After solving each task, the participants completed a NASA TLX questionnaire to evaluate the perceived cognitive demand of the tasks. Upon completing the study, participants were asked to express their likes and dislikes about each interface and indicate their preference.

4.2. Results

Of the 16 participants, 15 successfully finished the study. One user accidentally selected an incorrect target class for a batch labeling action using cVIL, which led to a dramatic reduction in accuracy. We excluded this participant from further analysis (see Section 5 for our learnings from this case).

All remaining participants performed better using cVIL compared to iVIL. The median accuracy of the exported labels compared to the ground truth was 96.05% for iVIL, whereas it was 98.4% for cVIL, as can be seen in Figure 4, which is a statistically significant difference ($t(14) = 5.784$, $p < .001$).

When comparing the final accuracy (measured against the ground truth) in dependence on the number of instance labels, we observe that cVIL achieves considerably better accuracy with fewer labels, as can be seen in Figure 6. The solid lines represent a robust linear regression estimation of the results for each framework. cVIL achieves around the same accuracy with 100 labels as iVIL with
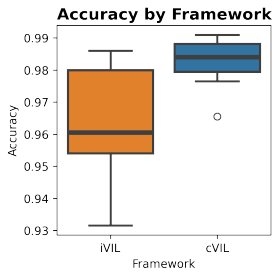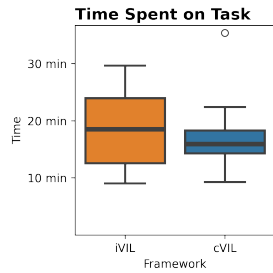
Figure 4: *Final accuracy.*
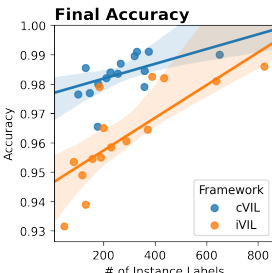

Figure 5: *Completion time.*

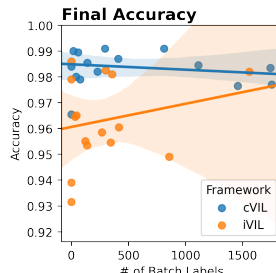**Figure 6:** *Final accuracy by number of instance labels.*

Figure 7: *Final accuracy by number of batch labels.*

600. Notably, these results were achieved solely through instance labeling, as can be seen in Figure 7. Interestingly, the accuracy is not significantly affected by the number of generated batch labels in both conditions (except for one outlier in iVIL). Participants batch-labeled an average of 600 samples in cVIL and 315 in iVIL, but this difference in the number of labels is not statistically significant ($t(14) = 1.570$, $p = 0.14$).

We also measured the time it took to finish the labeling task by looking at the difference between the first and the last labeling action or model retraining. Participants generally needed less time to finish the labeling tasks in cVIL as can be seen in Figure 5. The median labeling time for iVIL was around 18:30 minutes compared to 16 minutes for cVIL, however, this difference is not statistically significant ($t(14) = -1.947$, $p = 0.07$).

To analyze task load, we aggregated the scores from the NASA TLX questionnaire as described by Rubio et al. [RDMP04], where we weighted Frustration highest, followed by Mental Demand, Effort, Performance, and finally Temporal Demand. We could detect no difference between cVIL and the baseline (Wilcoxon Signed-Rank test: $z = -0.879$; $p = 0.39$).

In the final questionnaire, however, 13 out of 16 participants stated that they preferred cVIL over the iVIL baseline implementation. The main reason for this was that participants found the system easier to use as it helped them to better spot false positives and outliers, evaluate current model performance, and re-label incorrect predictions. Having information about uncertainty made it easier to label and gave a better indication of the model's accuracy. Participants reported that the class partitioning made it easier to focus on one particular class, and participants only needed to identify false

positives when deciding if a class label was correct, rather than having to consider the possibility of having true and false negatives as in an instance-based approach. However, participants disliked that there was too little change in the visual representations after the model updates. Four participants reported that the iVIL baseline was also easier to understand and navigate as well as more engaging and fun to use. However, six other participants felt it was more tedious and ambiguous since images were harder to find.

5. Discussion and Conclusions

In our study, users could label 2,000 photos in a reasonable time and with high accuracy with both interfaces. All users could achieve higher performance using cVIL, and most participants preferred it over the instance-based baseline. This confirms that class-centric VIL is promising when data is getting more complex – even though the data used for our user study was still relatively small. We conclude that scatterplots provide a less reliable way to judge model performance when class distributions strongly overlap compared to our class-centric property measure distribution plots. We found that cVIL can address these limitations and provide effective feedback with large and complex datasets while still being fully interactive.

Our study showed several directions for improvements of cVIL. In our study, we had to exclude one user because of an incorrect target class selection for batch labeling, which had drastic consequences on model accuracy. This could have happened with both interfaces and could be easily prevented for cVIL by warning the user if the selected target class does not correspond to the class density plot from which the selection was made. Generally, batch labeling using Min-Margin as the property measure (and also lasso selections in the baseline) did not notably contribute to model accuracy. In our simulations, data-based property measures produced more promising results for batch labeling. In the future, the usage of different property measures for instance and batch labeling should be investigated and empirically evaluated. Both our simulations and the user study show that cVIL is a promising approach in cases where data complexity becomes high and sample sizes are large, as is typically the case in real-world scenarios. Therefore, focusing on one class at a time and enabling the user to switch between instance and batch labeling depending on the state of the learning process shows to be of central importance.

A number of open topics remain for future investigation, such as the choice of the most suitable property measure, investigating different batch labeling strategies (including methodology from semi-supervised learning), and systematically evaluating the impact of data set size, features, and class properties on instance- and class-centric VIL performance.

Acknowledgements

References

[ACKK14] AMERSHI S., CAKMAK M., KNOX W. B., KULESZA T.: Power to the people: The role of humans in interactive machine learning. *Ai Magazine 35*, 4 (2014), 105–120. 1

[BGTF21] BENATO B. C., GOMES J. F., TELEA A. C., FALCÃO A. X.: Semi-automatic data annotation guided by feature space projection. *Pattern Recognition 109* (2021), 107612. 2

[BHS*21] BERNARD J., HUTTER M., SEDLMAIR M., ZEPPELZAUER M., MUNZNER T.: A taxonomy of property measures to unify active learning and human-centered approaches to data labeling. *ACM Trans. Interact. Intell. Syst. 11* (2021), 20:1–20:42. 2

[BHZ*18] BERNARD J., HUTTER M., ZEPPELZAUER M., FELLNER D. W., SEDLMAIR M.: Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics 24* (2018), 298–308. 1, 2

[BT20] BEIL D., THEISSLER A.: Cluster-clean-label: an interactive machine learning approach for labeling high-dimensional data. *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction* (2020). 2

[BZSA18] BERNARD J., ZEPPELZAUER M., SEDLMAIR M., AIGNER W.: Vial: a unified process for visual interactive labeling. *The Visual Computer 34* (2018), 1189–1207. 2

[CBB*19] CHEGINI M., BERNARD J., BERGER P., SOURIN A., ANDREWS K., SCHRECK T.: Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Vis. Informatics 3* (2019), 9–17. 2

[CNL11] COATES A., NG A., LEE H.: An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (2011), JMLR Workshop and Conference Proceedings, pp. 215–223. 3

[CTBA*11] CHEN M., TREFETHEN A. E., BAÑARES-ALCÁNTARA R., JIROTKA M., COECKE B., ERTL T., SCHMIDT A.: From data analysis and visualization to causality discovery. *Computer 44* (2011), 84–87. 2

[CTM*21] CARON M., TOUVRON H., MISRA I., J'EGOU H., MAIRAL J., BOJANOWSKI P., JOULIN A.: Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 9630–9640. 2, 3, 4

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), 248–255. 3

[FZL12] FU Y., ZHU X., LI B.: A survey on instance selection for active learning. *Knowledge and Information Systems 35* (2012), 249–283. 1

[GCT22] GRIMMEISEN B., CHEGINI M., THEISSLER A.: Visgil: machine learning-based visual guidance for interactive labeling. *The Visual Computer 39* (2022), 5097 – 5119. 2

[LCB10] LECUN Y., CORTES C., BURGES C.: MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist 2* (2010). 3

[LLWT15] LIU Z., LUO P., WANG X., TANG X.: Deep learning face attributes in the wild. In *ICCV* (2015), IEEE Computer Society, pp. 3730–3738. 2

[NA18] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics 25*, 8 (2018), 2650–2673. 2

[RDMP04] RUBIO S., DIAZ E. M. C., MARTIN J., PUENTE J. M.: Evaluation of subjective mental workload: A comparison of swat, nasa-tlx, and workload profile methods. *Applied Psychology 53* (2004), 61–86. 5

[Set09] SETTLES B.: *Active learning literature survey*. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009. 1

[SG10] SEIFERT C., GRANITZER M.: User-based active learning. *2010 IEEE International Conference on Data Mining Workshops* (2010), 418–425. 2

[SKK20] SCHRÖDER S.-M., KIKO R., KOCH R.: Morphocluster: Efficient annotation of plankton images by clustering. *Sensors 20*, 11 (2020), 3060. 2

[vdMH08] VAN DER MAATEN L., HINTON G. E.: Visualizing data using t-sne. *Journal of Machine Learning Research 9* (2008), 2579–2605. 2, 4

[ZTdJ*21] ZHANG Y., TENNEKES M., DE JONG T., CURIER L., COECKE B., CHEN M.: Simulation-based optimization of user interfaces for quality-assuring machine learning model predictions. *ACM Transactions on Interactive Intelligent Systems* (2021). 3