BI-LAVA: Biocuration with Hierarchical Image Labeling through Active Learning and Visual Analytics

Juan Trelles[1] and Andrew Wentzel[1] and William Berrios[2] and Hagit Shatkay[3] and G. Elisabeta Marai[1][2][†]

[1]University of Illinois Chicago, USA
[2]Universidad Nacional de Ingenieria, Peru
[3] University of Delaware, USA

Abstract
In the biomedical domain, taxonomies organize the acquisition modalities of scientific images in hierarchical structures. Such taxonomies leverage large sets of correct image labels and provide essential information about the importance of a scientific publication, which could then be used in biocuration tasks. However, the hierarchical nature of the labels, the overhead of processing images, the absence or incompleteness of labeled data, and the expertise required to label this type of data impede the creation of useful datasets for biocuration. From a multi-year collaboration with biocurators and text-mining researchers, we derive an iterative visual analytics and active learning strategy to address these challenges. We implement this strategy in a system called BI-LAVA—Biocuration with Hierarchical Image Labeling through Active Learning and Visual Analytics. BI-LAVA leverages a small set of image labels, a hierarchical set of image classifiers, and active learning to help model builders deal with incomplete ground-truth labels, target a hierarchical taxonomy of image modalities, and classify a large pool of unlabeled images. BI-LAVA's front end uses custom encodings to represent data distributions, taxonomies, image projections, and neighborhoods of image thumbnails, which help model builders explore an unfamiliar image dataset and taxonomy and correct and generate labels. An evaluation with machine learning practitioners shows that our mixed human-machine approach successfully supports domain experts in understanding the characteristics of classes within the taxonomy, as well as validating and improving data quality in labeled and unlabeled collections.

CCS Concepts
*• **Human-centered computing** → Visual analytics; • **Computing methodologies** → Machine learning algorithms;*

1. Introduction

Labeled image datasets are required in multiple practical applications of supervised machine learning (ML), from autonomous driving [GZL*20], spambot detection [KKZE19] and medicine [SJZ21, Bai21] to the image-based retrieval and to the identification and classification of scientific publications into biomedical databases (biocuration) [TLA*21, TASM23, TLA*20]. Not surprisingly, labeling solutions are urgently needed, rapidly creating a big industry market forecasted to generate a revenue of more than $17 billion by 2030 [Gra23]. The most common labeling settings involve recognizing and labeling the content captured in photographs, which is often a task humans excel at, even without training (e.g., recognizing the content as a boy holding a toothbrush), and which the industry rewards with relative low-pay ($1 to $25 per task) [The23]. However, other labeling tasks, for example aimed at identifying a specialized image type, require advanced knowledge at the postgraduate level, and would result in a pay in excess of $300 per task.

For example, biocuration aims to organize into taxonomies the vast information published in the biomed field. Although biomed scientific publications provide a vast source of unlabeled images, identifying the exact image type and subtypes from such a taxonomy (e.g., Experimental –> Gel –> Northern blot) can typically be performed only by highly skilled individuals named biocurators.

In such settings, incremental ML strategies, including but not limited to active learning (AL) [BZSA18], can help reduce the labeling costs by identifying smaller image subsets that require human intervention [Set09] and then learning from these smaller sets. In addition, strategies can leverage high-confidence predictions on the unlabeled set to add more labels, even though they do not come from a human source—i.e., they are pseudo- or weak-labels. Finally, integrating such ML strategies with visual analytics (VA) can also facilitate identifying and selecting labeling candidates from unlabeled data pools, and discovering knowledge in unfamiliar datasets.

When it comes to practical applications like the curation of biomedical data, several challenges from model and user per-

---

spectives hinder the integration of VA and ML techniques for understanding and labeling images into classes. From a model perspective, unlike typical systems [LCL*19, XXX*19, CSV*18, BNR20, CWW*20] that handle a single classifier and a single, non-hierarchical classification scheme (i.e., flat taxonomy), biocuration fundamentally relies on hierarchical taxonomies that may require multiple classifiers. This domain requirement complicates both the ML solution and the VA approach, because available labeled data may not match the taxonomy at the required level of specification—e.g., an image labeled as *microscopy* does not indicate whether it belongs to the *light* or *fluorescence microscopy* subcategories, i.e., the provided label is incomplete. Further issues in data quality, such as mislabels, imbalanced distributions of classes, and a lack of representative samples, can affect the model's performance [YCY*20]. In other words, the ML + VA solution should be able to handle incomplete ground-truth labels for many image sets, at multiple hierarchical levels, and possibly multiple classifiers.

From the user perspective, non-domain experts such as model builders may lack the expertise to correct and label domain-specific images. For example, classifying traffic light pictures is more manageable than distinguishing between *gel* and *plate* figures. Furthermore, although popular labeling tools such as Label Studio [TMHL20] or napari [nap19] can process a single image at a time, they lack the features necessary to process image collections, as in biocuration. Overall, model builders spend more time working with data than with models [Ana22], yet they have limited tools to understand unfamiliar image collections. A VA solution should support understanding such collections, because this type of support is crucial for labeling data efficiently and improving models' performance.

In this paper, we introduce BI-LAVA (**B**iocuration with **Hi**erarchical **I**mage **L**abeling through **A**ctive **L**earning and **V**isual **A**nalytics), a system that integrates feedback from an incremental ML strategy to aid the hierarchical exploration, understanding, and labeling of an unfamiliar image dataset. BI-LAVA is the first system to support novice labelers working with unfamiliar datasets. BI-LAVA is the only system to date to handle hierarchical classification schemes. BI-LAVA scales to thousands of images, and in that process, deals with cluttering issues via spiral image layouts which capture the neighborhood projections from embedding space to 2D. Unlike any other labeling system, BI-LAVA's ML backbone enables further feedback to help labelers understand problems with the classification models in terms of the data used for training (including incomplete labels), testing, or validation, and low and high-confidence samples.

Beyond the specific system capabilities above: 1) We introduce, characterize and document the biomedical image labeling process, from the novel perspective of model builders who collaborate with domain experts in biocuration. 2) We design and build a novel VA labeling system (BI-LAVA) to support this process, and leverage custom encodings and interactions. 3) We report and discuss feedback and lessons learned from a BI-LAVA evaluation with ML practitioners working with an unfamiliar collection of biomedical images.

2. Related work

2.1. AL and visualization for image classification

While our ML backbone could leverage a number of different learning strategies, as long as they can identify low and high-probability samples, our current instantiation uses AL and Cost-Effective Active Learning (CEAL) [WZL*17]. CEAL considers samples with low entropy and high probability of being correctly classified as pseudo-labels which are used to train the model in future steps. Low-confidence, high entropy labels need to be acted on by human labelers. Other approaches include Core-Set [SS18], which looks for the best subset for AL. However, Core-set is not suitable for large-scale labeled datasets. BI-LAVA uses CEAL due to its inexpensive, simple approach for separating low and high probability predictions. CEAL also produced promising preliminary results on our biomedical dataset, although it could be easily replaced by alternative approaches.

Previous work on Visual Interactive Labeling (VIAL) has explored the integration of visualization to augment human-in-the-loop AL, [BZSA18, SSZ*17, BHZ*18]. Other work has identified a taxonomy of data types and tasks in VIAL [BHS*21] with single views. We expand on this by exploring tasks for multiple linked views. Other works have explored multi-instance labeling with self-organizing maps [MBS*10] and pixel averages [HNH*12] to speed up labeling. In contrast, we focus on exploring and labeling the most similar images around an image of interest (i.e., image neighborhood).

Beyond AL, visualization research can generally aid in labeling image data. Several visualization systems attempt to detect and correct labeling errors [LCL*19, XXX*19, BNR20], identify a lack of representation in the dataset [CSV*18], and leverage unlabeled data [CWW*20]. Visual feedback also enables understanding model performance [RFT18, BHS*21]. Our work considers a more complex combination of these goals. Specifically, we target correcting label errors and analyzing labeled and unlabeled images, and we provide visual support for understanding data collections and model behavior. Furthermore, although most of these works employ AL, none deal with hierarchical taxonomies connected to multiple classifiers.

Two other works deal with related problems, but not for image data. VIANA [SSKEA19] enables annotating argumentation data using scrollable interactions to transition between different levels of aggregation, similar to our hierarchical taxonomy. However, unlike VIANA, we must always display the image labels and rely on a multiple-view system for annotating data and encouraging familiarization with the dataset. A second system, VASSL [KKZE19], uses multiple views to identify spambots by focusing exploration around an item of interest and filtering over dimensionally reduced elements. However, VASSL focuses on binary labeling. Still, both VIANA and VASSL are good examples of the need for contextual data for labeling, a current limitation in labeling systems like Label Studio [TMHL20] or napari [nap19].

### 2.2. Exploring sets or collections of images

Dimensionality reduction algorithms enable viewing and exploring large unstructured sets and collections of images by transforming

features derived from colors or latent representations to two dimensions. Markers and thumbnails on scatterplots are commonly used to display images [NW08], but they quickly clutter due to limits of display sizes. Alternatives to deal with cluttering include hierarchical clustering with zooming [CYL*21, XXX*19], scatterplots supported by grid-layouts to display thumbnails [XXX*19, LCL*19, CWW*20], or density contours to summarize data distributions and display a few samples [MG13]. Our design combines techniques from density contours with filters, selections, and supporting views to organize thumbnails.

Complementary techniques focus on placing thumbnails on a canvas. Grid-layout algorithms transform image positions from a projected space to a grid layout while preserving neighborhood similarities [CMC*21, FDH*15, DSF*14, BHA*22, GNCM*16]. While static treemaps may hurt thumbnail visibility [GCB*15], proposed variations support image exploration enabling interactions [BHA*22] or inspecting neighborhoods around an image of interest [WRZ*15]. Further techniques allow repositioning image projections [AA20] and summarizing groups [LZC*20] but are limited to exploring a small subset. As BI-LAVA follows the rationale of exploring images of interest based on static positions, we draw inspiration from spiral treemaps layouts [WRZ*15] for displaying thumbnails.

2.3. Hierarchy visualizations

Taxonomies of image modalities follow a hierarchical structure with different depths per branch. Typical encodings for this type of hierarchical data include indented lists [DLSP18], node-link diagrams (e.g., trees), icicles [KL83, vdWKB20] and treemaps [Shn92]. However, juxtaposed encodings are required to add further details [VBW15]. To aim for a compact representation of our taxonomy and distribution of samples, we use a combination of indented lists and horizontal bar charts, similar to work done for confusion matrices [GHM*22].
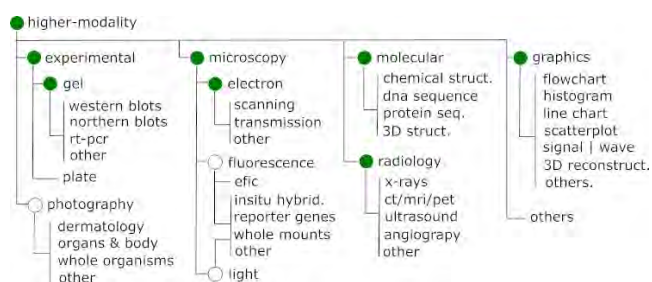
Figure 1: *Taxonomy of image modalities for biomedical images in COVID-19 papers [TLA*21, TASM23]. Nodes indicate parent classes with image classifiers; filled nodes further denote available classifiers at the start of the project. Nodes with blank fills denote expected classifiers where we did not have labeled data. For gels, light and fluorescence, sub-classes denote experimental methods.*

3. Application background

Unlike typical data labeling problems, we describe the challenge of unfamiliar domains, the issue of hierarchical incomplete labels,

and specific tasks related to the intrinsic need for hierarchical taxonomies of labels and multiple classifiers.

Biocurators extract knowledge from biomedical publications to populate scientific databases, saving an immeasurable amount of time to fellow researchers [Int18]. While text-based ML models accelerate biocuration tasks, such as document classification [JLK*20, BLP19], work from our collaborators found that complementing these models with image acquisition modality information produces better results [LJZ*21, SCB06]. Such results are consistent with the claims of how important images are for communicating scientific content [YAJC09, SCB06, CWM03]. However, integrating these modalities is underexplored due to the lack of standardization of classification schemes and the consequent lack of labeled data to train supporting image classifiers.

Modern attempts to automatically classify biomedical images into taxonomies use deep learning models [GSdHSBM16]. However, these taxonomies are not detailed at the level of granularity desired in biocuration. For example, they often classify images that use microscopy, but do not consider the modality (e.g., electron vs light microscopy) or sub-modality (e.g., scanning vs transmission microscopy), which has important information on the kind of experiments performed. To support creating more granular labels from sources with few or incomplete labels, we developed a hierarchical set of image classifiers following the parent nodes of a taxonomy developed for biocuration [TLA*21, TASM23] (Fig. 1).

Over the last five years, we have collaborated remotely with biocurators and text-mining researchers at three sites (Caltech, Jackson Labs, and Delaware) to integrate image and textual features to aid biocuration. Our team comprises two visual computing researchers and a senior undergraduate in ML. Some of our goals include harvesting labels for biomedical images and training image classifiers for image modalities, which lead to us exploring approaches for improving data labeling and the identification of modalities relevant to biocuration [TLA*21]. Through a series of semi-structured interviews, repeated observation, and feedback meetings, we became familiar with the existing labeling workflow and the image modality taxonomy to be used for classification.

4. Methods

Unlike other systems, our solution leverages multiple views of images and supports the multiple classifiers required by hierarchical taxonomies. BI-LAVA furthermore supports exploring unfamiliar hierarchical datasets to understand model behaviors, identify label errors, evaluate the ML performance, and identify data needs.

Our solution uses an iterative workflow that progressively processes an unlabeled collection of images by leveraging trained image classifiers and user inputs (Fig. 2). Following the VIAL framework [BZSA18], BI-LAVA has three outputs: data, model, and visual analytics. Users interact with labeled and unlabeled images to generate labeled data, which triggers the model output as a hierarchical set of image classifiers. Finally, the visual analytics component amplifies the knowledge output by enabling the understanding of the data and the model behavior.

To tackle the novel challenge of classifying image content at dif-

ferent levels of granularity, BI-LAVA uses a hierarchical image taxonomy. Unlike other systems that deal with a single classifier, we couple the taxonomy with a hierarchy of image classifiers where each classifier is a parent node in the taxonomy. To address the problem of incomplete or missing labels at different levels of granularity, BI-LAVA leverages a human-in-the-loop AL pipeline. This strategy allowed us to use each labeled sample, even when incomplete. For example, an image labeled as microscopy is used by the top classifier, while an image labeled as microscopy.fluorescence is used by the top classifier AND the microscopy classifier. Furthermore, this pipeline allows us to identify and correct issues in both the data and model. By allowing for an exploration of the dataset with the front-end VA module, BI-LAVA further addresses the challenges of working with unfamiliar datasets while improving data quality.

4.1. Data abstraction

Our system relies on data from images and the hierarchical information of the taxonomy of modalities. Most images were extracted from scientific publications from PubMed Central after we extracted and segmented subfigures. In addition to using the raw image data to train the classifiers and display thumbnails, the image schema includes the data source, split set type (training, validation, test, unlabeled), and label and captions (although not available at subfigure level, i.e., sub-caption), if available. We further derive latent representations, predictions, and associated probabilities using deep learning models. Other attributes include projected coordinates to 2D and metrics for the AL strategy (margin sampling [SDW01] and entropy [Sha48]). Our taxonomy of modalities is a tree structure where each child node increments the category's detail. Also, each parent node is associated with an image classifier and its performance metrics.

4.2. Activity and task analysis

We formulated the activities and tasks from our experience designing a labeling interface for biocuration [TLA*21] and from our attempts to find datasets that matched the specified taxonomy. Our previous labeling interface allowed biocurators to label images within publications on a one-publication-at-a-time basis, imitating their curatorial process. Although our system reduced the labeling time, selecting publications to label made it challenging to identify under-represented samples, leading to imbalanced datasets. Our initial dataset required merging images gathered from selected publications, competitions [GSdHSBM16], synthetic charts [Ado19], and other biomedical sources [DFAST12] in addition to a set of images labeled by curators. This laborious process was time-consuming because biomedical images required extra time to verify our labeling decisions. As model builders, we needed more domain knowledge to discern between modalities easily.

After the start of the COVID-19 pandemic, our collaborators wished to shift our image-harvesting efforts from biomedical papers to a growing collection of COVID-19 publications [WLC*20]. However, our manual labeling interface did not scale well to the much larger number of COVID-19 documents (more than

250,000). Our group could also not afford the time to manually inspect samples to validate labels and image quality. Instead, we aimed to leverage the image classifiers we had developed for the much smaller set of labeled images and leverage our experience as model builders to replace biocurators as data annotators. However, as we approached the new classification problem and researched possible solutions, we learned that other model builders faced similar problems across domains related to the same lack of domain expertise with a dataset. Consequently, providing visual cues to support data understanding before labeling became essential.

We documented the issues encountered by our group and problems reported by other engineers with similar issues, and conducted semi-formal interviews with several ML practitioners and with our biocurator collaborators. By further contrasting the resulting needs with tasks characterized for instance selection [BHS*21] and supported by visual encodings [SG17] (both discussed in the Related Work 2.1, 2.2), we arrived at the following list of activities and tasks [Mar17]:

***A1**. Analyze labeled and unlabeled images for each class defined by the hierarchical taxonomy*

- T1.1 Display an overview of the labeled samples and their relationship to the image classifiers used in the taxonomy
- T1.2 Show images by similarity and allow the comparison between labeled and unlabeled data subsets
- T1.3 Explore neighborhoods of images to become more familiar with data characteristics and characteristics shared between similar images
- T1.4 Identify how model behavior changes for different types of images, and what features commonly lead to labeling errors

***A2**. Select candidates to update and label images*

- T2.1 Identify labeling errors in the labeled and unlabeled sets corresponding to each image classifier
- T2.2 Update image labels for single or multiple images
- T2.3 Browse low-confidence samples to identify labels that require human-intervention
- T2.4 Browse high-confidence samples to support the evaluation of the pseudo-labels
- T2.5 Evaluate modifications to data labels

Beyond general exploratory labeling (T1.1, T2.1 [BHS*21]), T1.2 - T1.4 are specified for our project's labeling needs, although they resemble tasks typically supported by scatterplots [SG17]. T2.3 - T2.4 are unique to our domain problem [BHS*21]. Non-functional requirements included handling large image datasets of up to 500,000 images while allowing an interactive browser experience and supporting on-demand calculations of dimensionality-reduced features (a computationally-expensive operation). Finally, our system should be easily usable for model builders with limited visual literacy.

4.3. Pre-labeling and image classifiers

Pre-labeling refers to annotating data offline before training the classifiers [BZSA18]. We obtained labeled data from a few labeled datasets, including ImageCLEF 2013 and 2015 [GSdHSBM16],
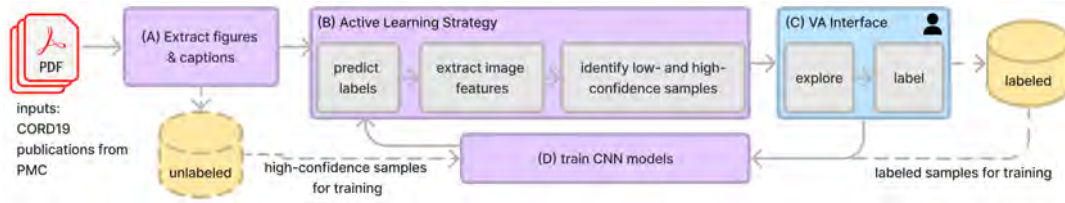
Figure 2: *Architecture of BI-LAVA's active learning system. (a) An offline pipeline extracts images and captions from PMC COVID-19 related publications. (b) The AL strategy uses previously trained CNN models to predict labels, and extracts image features from unlabeled images by fetching latent representations from the last convolutional layer. Next, it separates images based on the classifier's confidence according to two popular metrics. (c) A user accesses the visual analytics interface to understand the collection of images and label them. (d) Labeled images and high-confidence unlabeled samples are used to retrain the image classifiers.*

from approximately 6,000 data labeled by biocurators on our previous labeling interface [TLA*21], and from a synthetic charts dataset [Ado19]. In addition, we scrapped around 2,000 images from Open-i [DFAST12] and added approximately 15,000 samples of *experimental* images provided by our collaborators. Except for the synthetic dataset for charts, all the other images come from scientific publications that guarantee verifiable provenance to a biomedical source. We note that, in contrast, searching the web for 'light microscopy' returns generic images of microscopes instead of the expected light microscopy images. Our training efforts focused on a subset of classifiers due to labeled data availability (green nodes in Fig. 1). We made large efforts to guarantee that the data was suitable for training; however, we could not guarantee that mislabeling did not happen. In addition, we disregarded samples labeled by biocurators that included extraction errors (e.g., wrong segmentations in Fig. 6g) generated while extracting content from PDF documents. We manually inspected the samples for every labeled dataset and matched them to our taxonomy. Our total pool of labeled images contained 333,998 samples.

We trained each parent node of the taxonomy as a supervised classifier independently where the classes were the node's children. To keep the training sets consistent across parent and child classifiers, we first trained the lower levels of the hierarchy using a stratified partition with training, validation, and test partitions of 70/10/20. For the parent classifiers, we follow a similar approach but also force images previously selected for a training set to remain in the parent's training set and avoid data leakage. For instance, samples in the training dataset of the *electron microscopy* classifier also belonged to the training dataset of the *microscopy* classifier. Given its competitive results, our chosen architecture was a ResNet18 [HZRS16] model. We trained these models using transfer learning from ImageNet and early stopping to avoid overfitting, with a learning rate of $1e-4$, and saved the F1 scores.

Given our collaborators' requirements for gathering COVID-19-related images, we created our unlabeled dataset from the CORD19 dataset [WLC*20]. This dataset contains entries to research articles about COVID-19, updated periodically until recently. We extracted approximately 32,000 documents from that subset available before January 2021 from PubMed Central. As the CORD19 dataset does not provide images, we used the PMC identifiers to collect

the publications as PDFs using NIH's interfaces. We detail the preprocessing steps next.

4.4. Active learning strategy

We obtained the biomedical images for the unlabeled dataset by pre-processing publications in PDF format. First, we pass every document through an extraction pipeline [TLA*21] to obtain every figure, subfigure, and corresponding caption (Fig. 2a). The pipeline internally invokes modules for image and captions extraction [LJKS18], and figure separation [LJS19]. We follow this approach as figures in biomedical publications can contain subfigures with different modalities that would add noise to the training data. We then store these images in our unlabeled database.

Next, we start our AL strategy on the unlabeled collection of images (Fig. 2b). Using the pre-trained models, we infer image labels and prediction probabilities and extract the image features (from the last convolutional layer in the model before the softmax layer). We save the image features for the dimensionality reduction step later in the workflow. This step is repeated per classifier in the taxonomy from top to bottom following a branch of the taxonomy in order to treat each modality independently. High-confidence samples are identified in the dataset as images with an entropy [Sha48] below < 0.05, based on the CEAL framework [WZL*17], which are used as pseudo-labels. Images with lower confidence are ranked using the margin sampling score [SDW01] to provide more information to prioritize labeling decisions.

The following step in the workflow involves a model builder interacting with our visual analytics interface (Fig. 2c). We describe the components in Sec. 4.5. In this stage, the model builder produces new labels by confirming or correcting predicted labels. Confirmed pseudo-labels become training data in the AL approach. Finally, the training step (Fig. 2d) uses labeled data and high-confidence samples as a training pool to retrain the models from scratch before reinvoking the AL steps. Because training the model takes a considerable amount of time, we only retrain the model when training pools for a classifier, including new or updated labeled samples, are incremented by a user-defined threshold.
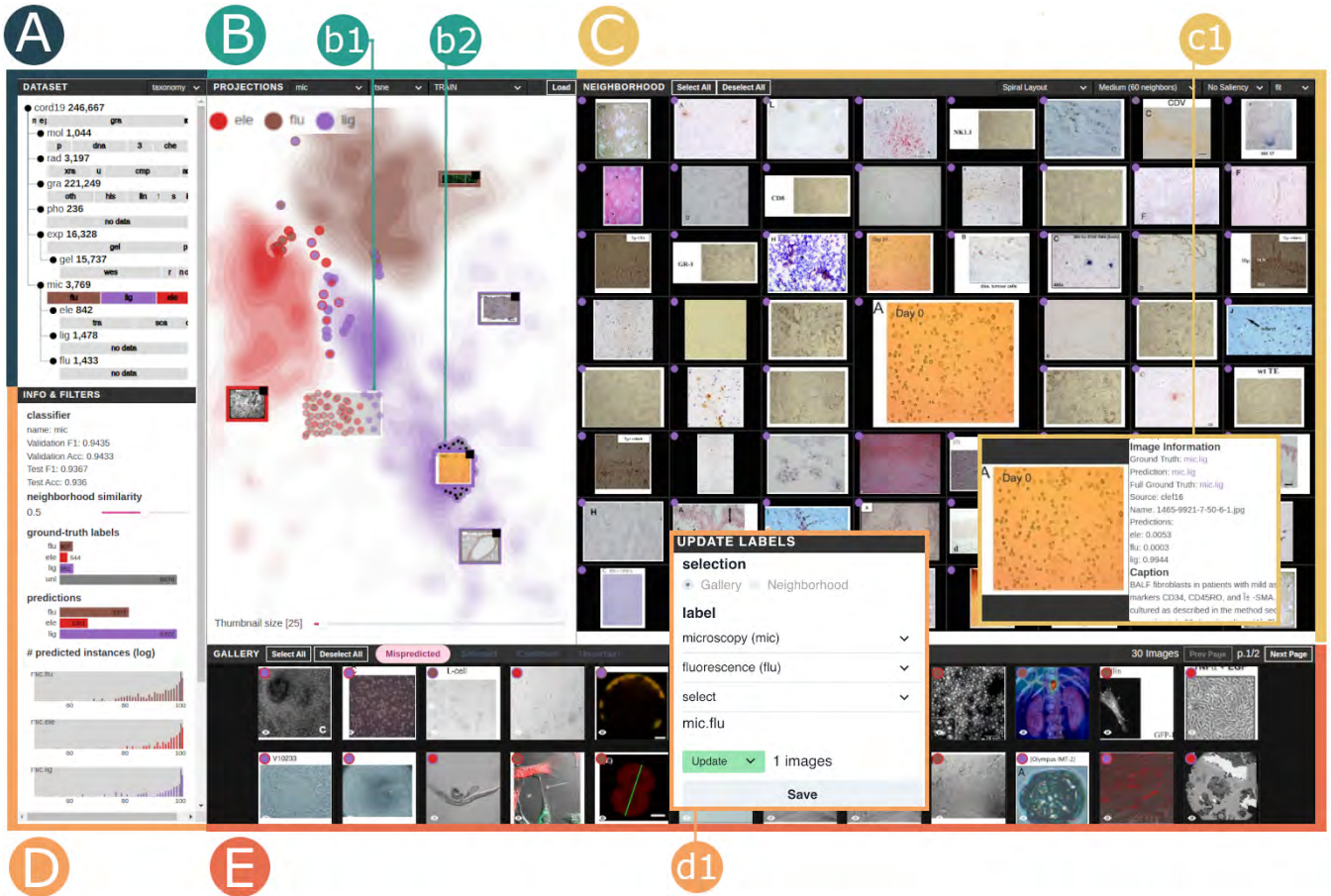
Figure 3: *Visual interactive labeling with BI-LAVA. (a) Dataset view displaying a taxonomy of image modalities and proportion of samples per class. (b) Projection of a training set of microscopy images using t-SNE. Density heatmaps hide confident samples, while circle encodings show samples with a neighborhood similarity below 50%. Brushing reveals samples in the lower region (b1). (c) Neighborhood view displaying an image of interest from the scatterplot (b2) and neighbors in a spiral layout. Details on demand show the image caption and prediction probabilities (c1). (d) Performance metrics from classifiers and filters. Scrolling down reveals the Update panel (d1 positioned on the side for readability). (e) Gallery of images for different data subsets: mispredicted samples, samples from the selected area, confusing samples, and uncertain samples.*

4.5. Front-end design

We designed our visual analytics platform to address the previously discussed challenges of model builders when dealing with unfamiliar datasets, ultimately leading to more efficient labeling. BI-LAVA (Fig. 3) provides different views and interactions to support data understanding and labeling. The *Dataset view* (a) provides an overview of the distribution of labeled samples per class in the taxonomy (*T1.1*) and an overview of the current label updates (Fig. 4) in a user session (*T2.5*). The *Projection view* (b) allows the exploration of projections from image features (*T1.2-1.3, T2.1*) and inferring model behavior (*T1.4*). Users can filter the data using the panel to its left (d). The *Gallery view* (e) provides different entry points to images (*T1.3,T2.3,T2.1,T2.4*). The *Neighborhood view* (c) enables exploring the most similar thumbnails (*T1.3-1.4*) and explains predictions (*T1.4*). Finally, users can update labels (*T2.2*)(d1). This design results from a parallel prototyping ap-

proach [DGK*10] with feedback from three visual computing researchers. We implemented the front end using React, D3 [BOH11] and ThreeJS, and used Flask, PyTorch, RAPIDS [RAP23], and MongoDB for the back end.

4.5.1. Dataset view.

The *Dataset view* provides an overview of the labeled data and labeling updates performed in a session. Visual cues attempt to guide model builders' exploration, such as showing classifiers with imbalanced distributions of labeled data. After selecting a node in the taxonomy from the top left of the interface, the view for labeled data summarizes the distribution of labeled images per category as an indented list (Fig. 3a) with the total number of samples per category. In this tree-like structure, each node represents an image classifier. Below, a horizontal bar takes the whole available width space to display the distribution of images per category (*T1.1*). Conse-
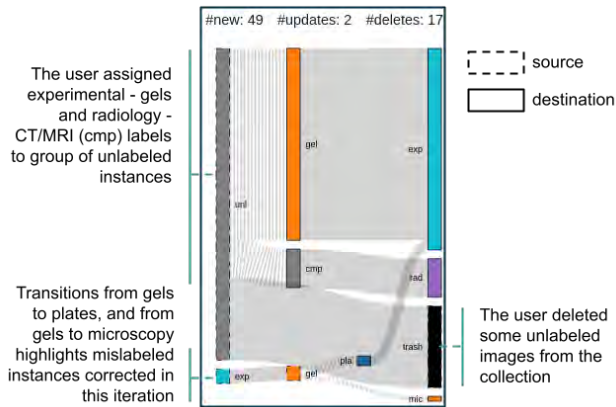
Figure 4: *An alternative view in Dataset view (Sec. 4.5.1) for displaying labeling updates during a user session.*

quently, the bars are not comparable between classifiers. We made this design decision as our exploration focuses on one image classifier at a time. Bars are color-coded following a chromatic scale of 10 different colors. However, given the large number of nodes in the taxonomy, some categories may share the same color across different classifiers.

Design alternatives included trees and treemaps. However, juxtaposing the taxonomy and image distribution details provided a more compact and understandable representation. In contrast, treemaps produced small squares due to the imbalanced number of images between categories. Zoomable treemaps were another alternative, but they required more interactions and did not provide an overview of all nodes.

Estimating the number of labels to update before finishing a labeling session is a hard challenge. Few images may lead to no significant changes, while many label updates demand more participation from human annotators. Instead of recommending such a number, BI-LAVA displays an overview of the changes when the user selects the updates option in the toolbar. A Sankey diagram (Fig. 4) summarizes the label updates done so far in the labeling session, including new labels, updated labels, and deleted items (*T2.5*). Boxes to the left (with dashed borders) represent the taxonomy categories affected by the updates, while boxes to the right (with solid borders) represent the updated values. A black box complements the boxes to the right as a placeholder for deleted elements. Our representation includes any parent category for an updated taxonomy. These update flows in the view provide cues for common mistakes in the current dataset, such as labeled samples moved to another distribution (i.e., mislabeled).

4.5.2. Projection view

The *Projection view* displays the images associated with an image classifier on a 2D scatterplot (Fig. 3b). Due to the documented benefits for novice users, we followed the "overview first, zoom and filter, and details on demand" mantra [Shn96] to support data understanding. Toolbar options enable selecting the classifier, projection technique, and dataset sub-set. BI-LAVA calculates the dimen-

sionality reduced features to 2D using either PCA, t-SNE [vH08] or UMAP [MHM20], leveraging the RAPIDS GPU-based implementation to obtain acceptable on-demand processing times (e.g, from 5.8s for 50,000 images to 3m for 500,000 images, using t-SNE). The dataset sub-set options include training, validation, test, unlabeled, unlabeled + train, or all the data. These options allow the user to narrow down the dataset based on their intent (*T1.2-T1.4*). For example, training samples include ground-truth labels and high-confidence, which are more useful to get an initial idea of the data and whether data expected to be correct has labeling issues or erroneous predictions (*T2.1*). In contrast, unlabeled + train shows how the unlabeled elements relate to the training pool.

We use a combination of a scatterplot and density heatmaps to represent the image data. Given the large dataset size, we first show an overview of the data by displaying the different clusters of images as density heatmaps. Similar to Splatterplot [MG13], we then display images of interest on top of the heatmaps. Since we are interested in images that are more likely to be mislabeled (*T2.1*), we show images whose neighborhood hit metric [PNML08] (percentage of neighbors with the same label) is below a user-defined threshold. The rationale behind this decision is that users may be interested in exploring first images surrounded by different neighbors, which usually lie on cluster borders. Users can manipulate this threshold at will (Sec. 4.5.3). We encode each sample as a circle where the background color represents the ground truth, and the border color represents the prediction label, following the color scheme used in the *Dataset view*. For unlabeled samples, we fill the circle background in gray. Image thumbnails appear as details on demand after clicking on a circle mark.

Analyzing these sub-sets enables understanding model performance and problematic regions [RFT18]. Thus, this view depends on the notion of similarity obtained in the 2D space to guide exploration. However, these projections could suffer from distortion errors [NA18]. The *Neighborhood view* mitigates this effect by showing the raw image data (Fig. 3c).

This view supports interactions to ease exploration and is linked to other views in the interface. First, users can zoom into an area with mispredicted samples to inspect neighbors and reveal their thumbnails by clicking on a circle mark. Then, clicking on the top right of the thumbnail triggers the *Neighborhood view* (Sec. 4.5.5). Next, by brushing over a region, the view reveals the samples hidden due to the neighborhood hit threshold (Fig. 3b1). This image selection is also displayed on the *Gallery view* (Sec. 4.5.4) as a grid layout of thumbnails. At last, the *Filters view* allows filtering of the view by image metadata (Sec. 4.5.3).

4.5.3. Filters view

BI-LAVA leverages image metadata to filter an image collection (*T1.2*). Bar charts and histograms summarize the values of the current data sub-set, including unlabeled data if applicable. We use bar charts to show the distribution of the ground truth labels, predictions per class, and data sources. Each bar in the charts acts as a filter, and updates to the bar fill indicate the active filters. Users can identify discrepancies in model behavior by combining these filters. For example, filtering electron microscopy images should

ideally display only images predicted as that category. Hence any other misprediction is worth reviewing.

For the model predictions, we use a histogram for each class to show the log distribution of samples for each probability bin. We chose a log distribution to account for the imbalanced distribution of samples. Two color-coded bars are displayed side by side per prediction probability, one for labeled and one for unlabeled samples (*T1.2*); this setup allowed us to gain vertical space. Sliders filter the values. For example, we can filter samples with prediction probabilities lower than 60%. Finally, the top area displays the classifier's performance metrics.

4.5.4. Gallery view

The *Gallery view* (Fig. 3e) complements data exploration by showing images of interest based on user interaction, classification outputs, and outcomes of the AL strategy. Content is organized using tabs. The first tab groups the mispredicted images. For example, discrepancies between ground truth and predicted labels indicate potential issues in the classifier (*T2.1*). The second tab groups selected images from the *Projection view*. Finally, the last two tabs group high-confidence samples, starting with the most confident images, and low-confidence samples, starting with the most uncertain images, respectively (*T2.4,T2.3*).

This view places these images using a paginated grid to show the image content. We place a circle mark for each image on the top left, following the same color-coding convention as in the *Projection view*. In addition, a red circle mark on the top right indicates an image marked for deletion, while a color-coded circle mark indicates that the image label has been changed to the color-coded label. As images in our dataset have different aspect ratios, we chose squared elements to display and fit their content to the constrained space. However, hovering over a thumbnail expands the image content.

As an alternative to the "overview first" approach in the *Projection View*, the gallery enables a "details-first, show context, overview last" approach [LBS*18]. Model builders can click on an image in the gallery to reveal its context, depicted as a neighborhood of thumbnails (*Neighborhood view*), and later inspect the neighborhood's location in the whole projection space. For instance, *unknown unknowns* [CSV*18] appears when a sample with high confidence is mispredicted. If users spot one of these cases, they can inspect the neighborhood for more potential errors. Finally, selecting thumbnails indicate labeling candidates (Sec. 4.5.6).

4.5.5. Neighborhood view

While the projection view provides flexibility for viewing, filtering, and selecting regions in the dataset, visualizing image thumbnails is challenging due to clutter. We overcome this problem by inspecting image thumbnails on a separate view from the perspective of an image of interest. This rationale allows expanding the exploration from an image sample presented on the *Gallery view* or found through investigation on the *Projection view*. For example, an uncertain image can reveal more patterns in a neighborhood that may confuse the model. In particular, the *Neighborhood*

view (Fig. 3c) sorts the neighboring images based on their proximity (i.e., similarity) to this image of interest using three layouts: a spiral layout [WRZ*15] (Fig. 3c), a novel variation of this spiral layout that preserves the relative position of the projected features in the 2D landscape (Fig. 6e,g), and a traditional grid layout. While grid layouts are more common than spiral layouts, we decided to use spirals as the default layout as they highlight the image of interest by placing it in the center of the canvas. The local exploration in this view aids the inspection of visual features per region. Understanding these similarities helps model builders create a mental model of the classifier's behavior (*T1.3,T1.4*), like identifying regions with chest X-rays.

Spiral layout. The spiral layout for images [WRZ*15] places the image of interest in the center and neighbors in concentric rings. The order of the neighbors, sorted by distance, starts from the top-left quadrant of the central image and proceeds clockwise (Fig. 5b). We calculate the neighbors based on the Euclidean distance in the 2D projected view and mitigate distortion errors by displaying the thumbnails in this spiral layout. In addition, the layout organization approximates the elements' position in 2D to provide consistency. Each ring contains thumbnails with the same aspect ratio as the central image to fit perfectly on every edge of the central element. If required, the method is flexible enough to consider further size reduction of outer rings by dividing the width and height in multiples of 2. As the spiral layout follows a treemap algorithm, the spiral pattern can recursively appear for boxes with hierarchical elements. However, as image sizes shrink fast, the recursion affects the visibility of the thumbnails. Thus, we restrict our design to divide each rectangle into four elements at most if required (Fig. 5c). We use the spiral layout as the default setting.

Spatial spiral layout. Although the spiral map arranges elements by distance to the central image, the position of each element does not preserve the structure of the elements in the 2D projection. Figure 5 displays an example case of this problem. The scatterplot (a) displays a set of random points, color-coded by the quadrant where they appeared, with the point of interest in the center. The spiral layout (b) uses the distance to the center to organize items in concentric rings. However, the resulting arrangement places items from the same quadrant far apart, violating the structure given by the scatterplot. Our alternative arrangement (c) organizes items by similarity while preserving quadrant membership. It should be noted that an unequal distribution of elements per quadrant requires a subdivision of the space to fit the boxes. These ideas lay the foundations for our *spatial spiral layout*.

To build a spatial spiral layout, first, we choose the number of rings in the layout. Next, we build each ring by subdividing a sorted list of points by quadrant until we fill in all the rings. When elements in a new ring have the same size as in the previous ring, the new ring allocates four more elements. However, when the outer ring halves the size of the elements, it contains double the elements plus 4. Then, we traverse the list of ring elements in reverse order and subdivide each element into four. Our *Neighborhood view* considers cases for 2, 3, 4, and 5 rings called small, medium, large, and very large views, respectively, that can allocate from 32 to 180 images (supp. materials).

Features and interactions. In contrast to the *Gallery view*,
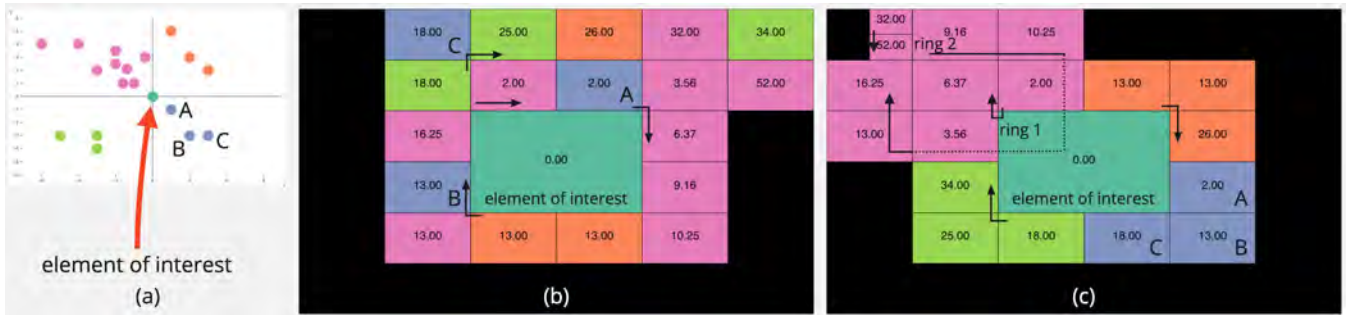
Figure 5: *Spiral layout vs Spatial spiral layout. (a) Random distribution of points in a 2D projected space, color-coded by Cartesian quadrant, surrounding a point of interest in the center. (b) Spiral layout with two concentric rings. Color encodes quadrants and unused space (black), and numbers show distances between an item and an element of interest. (c) Spatial spiral layout positioning points based on their quadrant and distance. The top-left quadrant requires subdivisions.*

scaled images match the rectangular space available in the layout. From previous experiences working with biomedical images [TLA*21], images with unusual aspect ratios (e.g., skinny images) may indicate the presence of extraction errors while processing PDF documents. Showing thumbnails covering the squared space, as in the *Gallery view*, is also available. Markers for labels and updated information are consistent with the design in *Gallery view*. Additional features support the exploration and actions in the neighborhood of thumbnails. To provide a proxy for the interpretability of our image classifiers, we provide saliency maps using GradCAM [SCD*17]. We color-encoded the saliency maps using red for areas with low activations and blue for areas with high activations. Users can see more details by clicking on markers, such as captions, data sources, and predicted probabilities per class (Fig. 3c1). Finally, as in the *Gallery view*, users can indicate the images to update by clicking on the thumbnails or using the toolbar options.

4.5.6. Update panel

This panel allows updating the selected samples from the *Projection* or *Gallery views* (*T2.2*). Users can indicate new labels or mark items for deletion (Fig. 3d1). We also allow specifying child labels or different parent nodes if the samples were misplaced (i.e., out-of-distribution).

5. Evaluation

Beyond manual labeling, which BI-LAVA partially automates and thus clearly outperforms, there are no other labeling systems that BI-LAVA could be reasonably compared quantitatively against. Instead, we evaluate BI-LAVA through two case studies with several practitioners and rigorous qualitative feedback. In the first case study, a senior model engineer with experience with biomedical images used BI-LAVA to analyze two subsets of images extracted from COVID-19 publications in conjunction with labeled sources. In the second case study, five ML practitioners independently used BI-LAVA on a subset of a biomedical image dataset, which was unfamiliar to them. In both case studies, we used a think-aloud protocol with note-taking, followed by a questionnaire.

5.1. COVID image modality labeling

This case study aimed to identify problematic instances from two types of imaging modalities in the COVID-19 dataset and update the labels accordingly. The study was completed by a senior model engineer working with the dataset as part of the biocuration project using a think-aloud protocol with note-taking. The engineer first sought to investigate the imbalanced distributions of samples he had used to train image classifiers—namely, the subset of gels and plates in the experimental (exp) assays category. Second, the engineer analyzed the molecular (mol) category, whose images are known for containing text strings such as RNA and DNA sequences. The experimental and molecular classes are handled by two different classifiers in the second level of the hierarchical taxonomy.

The engineer started his analysis by inspecting the *Dataset view* to locate a modality category with an imbalanced distribution of samples (*T1.1*). After noticing a significant disparity between the gel and plate sub-classes (15,737 gel vs. 591 plate samples) in the experimental node (Fig. 6a), he loaded the training and high-confidence unlabeled data using UMAP, due to previous experience with this technique (*T1.2*). Figure 6b shows one big cluster in the center dominated by gels (in orange) and two small clusters on the sides, mainly grouping samples of the same class (plates shown in blue). He preferred first to analyze the small clusters to determine the reason for the presence of few gels in a dominant plate region, or vice-versa.

By zooming in on the cluster on the far right (Fig. 6c), the engineer identified three samples with a ground-truth label of gel predicted as plates within a blue region (plates). Then, he clicked on the instances in the *Projection view* to visualize the thumbnails: two looked pretty similar, while the last one looked like an out-of-distribution sample (*T2.1*). Next, the engineer tried to brush over the area to open more thumbnails, but the view became cluttered. Therefore, he selected one suspicious sample and opened the *Neighborhood view* (Fig. 6d, *T1.3*). As most neighbors appear to the sample's right, he switched the view to the spatial spiral layout (Fig. 6e). *"It becomes clear now that this image is mislabeled as gel while it is a plate. They all share these circular patterns"*, he added.
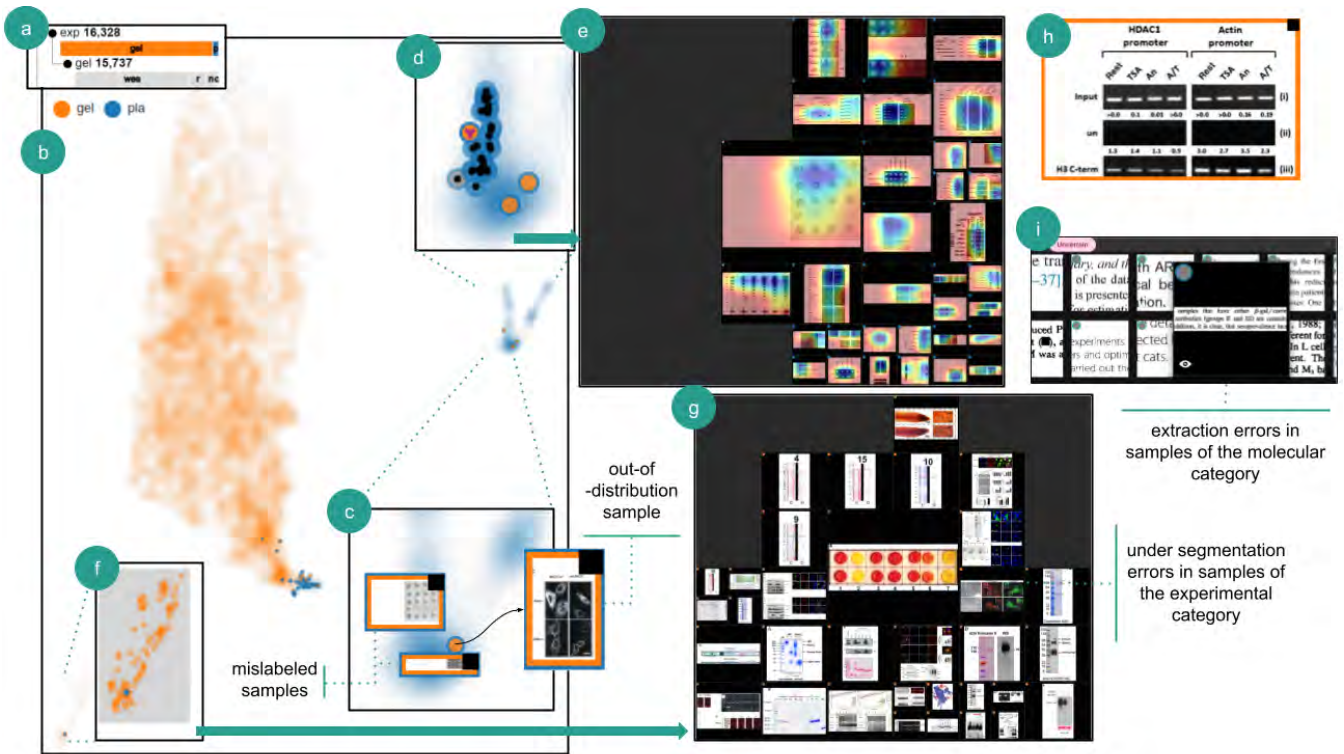
Figure 6: *Interaction during the case study: COVID-19 Image Modality Labeling. (a) Imbalanced distribution of samples for gels and plates. (b) Projected image features for gel (orange) and plate (blue) samples using UMAP. (c) Zoom in to the plate region on the right displaying the thumbnails for three samples labeled as gel but predicted as plate. The cluster contains an out-of-distribution sample. (d) Mispredicted image of interest and immediate neighborhood in the plate cluster. (e) Neighborhood view using a spatial spiral layout and saliency maps. Circular patterns yield higher activations (in blue). (f) A plate sample (blue) is predicted as gel (orange). (g) Neighborhood view for a plate image with unconventional colors in a gel region. Neighboring elements include out-of-distribution and under-segmented samples. (h) An experimental gel sample. (i) Uncertain samples in the molecular category representing portions of text.*

An inspection of the saliency map further corroborated that the circular patterns caused the highest activations in the model. The engineer then selected the other mispredicted sample, also a mislabeled instance, and updated the ground truth label to plate. The engineer commented on the remaining mispredicted gel sample: *"I have seen these images before when working with the microscopy classifier. It probably is a fluorescence sample, so it is peculiar to find it classified as a gel"* (Fig. 6c right). After opening the details for each of the three images of interest and checking the image source, he added: *"These three samples came from the same labeled collection we generated using a previous version of a classifier. I now see that we most likely got something wrong and need to revisit several labeled samples."* In addition, he identified several thumbnails with very narrow aspect ratios, which he confirmed were over-cropped figures that needed to be discarded (*T2.1*). He ended the cluster inspection by deleting the errors and moving the mislabeled sample to the microscopy set (*T2.2*).

Next, the engineer wished to check the isolated plate instance within the gel cluster on the left side (Fig. 6f). This sample looked like a plate but used a different color scheme (Fig. 6g). After brushing over the plate cluster previously explored, he hovered over the

selected tab in the *Gallery view*. He commented: *"For me, this image is a plate, but the color is different; no caption is available to be sure. Also, this gel cluster has many instances that should not be under the experimental category."* After filtering samples by source, he stated again that one of the labeled datasets needs to be revisited as it contains noisy samples, including images with multiple modalities (*T2.1*).

Then, the engineer focused on the remaining mispredictions and evaluated the predictions in the unlabeled dataset. He noticed that the issues appear in a tiny area in the bigger cluster's boundaries of gels and plates. Thus, he decided to open the *mispredicted* tab in the *Gallery view* to understand the issues in these 58 instances (*T2.1*). He quickly identified more microscopy images that should not have been in this dataset. Then, curious about the classifier's performance on the unlabeled dataset, he inspected the *confident* tab (*T2.4*). Approximately 95% of unlabeled samples were predicted as gels, and fewer than 1% of instances had issues. In contrast, plate predictions contained roughly 10% of out-of-distribution samples, such as radiology CT scans of chests and brains. He commented that these particular samples were problematic as the model classifies them as highly confident predictions. Based on this inspection

of the confident samples, he decided to show only unlabeled samples on the *Projection view*, brush over the dense gel regions in the middle, and open one unlabeled sample in the *Neighborhood view*. The image caption listed the term *western blot*, a subclass of gel. As all the images in the neighborhood looked quite similar, he displayed 180 neighbors and confirmed the label of the unlabeled samples as gel (*T2.2*). He finalized this inspection by mentioning that the 99% F1 score of the experimental classifier cannot be fully trusted as it hides many errors, such as mislabels and lack of representation.

Last, the engineer targeted the molecular dataset. He hypothesized that images containing text annotations could affect predictions for subclasses mainly containing characters, such as protein sequences. He started the analysis by plotting the unlabeled subset for the molecular (mol) category. Surprisingly, the *uncertain* tab did not show instances of confusing biomedical images containing text but instead showed small chunks of text extracted from paragraph regions in the PDF (Fig. 6i). He marked these images for deletion and commented: *"These samples indicate an error on our extraction algorithm. It would be good to mark them as an 'other' class within the molecular dataset as a proxy for these mistakes"*. In contrast, an inspection of the confident samples showed a good performance on chemical structures and DNA sequences, which appear as phylogenetic trees of nucleotide samples.

5.2. ML practitioners exploring unfamiliar data

This case study aimed to collect qualitative feedback about the benefits of BI-LAVA when an ML practitioner targets an unfamiliar dataset for data labeling. We recruited five ML practitioners to use BI-LAVA and perform a series of tasks to achieve this goal. Only one practitioner (P4) had previous experience with data visualization. Also, four practitioners were not affiliated with our project. Although we recognize the fifth practitioner (P5) as a co-author, his contributions focused on the AL components. We completed four in-person sessions and conducted the last session over Zoom and Parsec due to our remote collaboration. For in-person sessions, practitioners interacted with 31" monitors (1920x1080). For the remote session, the practitioner used a 15" monitor (1366x768). Every session lasted approximately 90 minutes and employed a think-aloud approach with note-taking.

At the beginning of each session, the practitioners read a description of the project background, and a facilitator addressed any additional questions. Then, the facilitator demonstrated the system features in one of the taxonomy classes. Next, the practitioners performed the following four tasks on the experimental assay (exp) dataset: (1) explore the dataset and identify characteristics for each class; (2) identify noisy samples; (3) identify high confidence samples from the unlabeled dataset to leverage as training samples; and (4) label low confidence samples from the unlabeled dataset. Finally, they provided qualitative feedback in an online questionnaire. During the session, the facilitator addressed any concerns. We chose the experimental assay as the unfamiliar dataset as it only contained two classes, and as shown in the previous case study, it had an imbalanced distribution of samples.

The first task aimed to familiarize practitioners with the dataset by exploring the samples used in the training dataset and the unlabeled samples with high confidence. t-SNE was the dimensionality reduction algorithm most used (4/5), in contrast to PCA and UMAP (P4). Different exploration strategies (*T1.2*) arose for this task, including sampling thumbnails across regions (similar to Fig. 3B), exploring regions far from the overlapping region, starting from an overlapping region, and starting with low-density regions. During this stage, few practitioners (P3, P4) used filters and preferred using the *Neighborhood view* (*T1.3*).

The most common approach to validate their hypothesis about the data was using the *Neighborhood view* to inspect neighboring thumbnails. Some practitioners (P1, P3, P4) also used a combination of the *Neighborhood*, *Gallery*, and *Projection views* by showing thumbnails in the scatterplot, selecting a distant region to display thumbnails on the gallery, and then contrasting those samples with the neighborhood. The spatial spiral layout was often preferred as it matched better the structure in the scatterplot. By the end of this task, practitioners noticed that circular patterns were prominent on plates (Fig. 6e) while stripes were more common on gels (Fig. 6h). A practitioner added that there was some grid organization between the elements in gels and plates.

For identifying noisy samples in the second task, practitioners followed different strategies (*T2.1*). Most practitioners (3/5) navigated to the overlapping regions or sought isolated samples on opposite clusters. For example, they identified a plate sample surrounded by gels (Fig. 6c). P2 preferred the *mispredictions* tab as a starting point for exploration, while P3 used filters to show gel labels predicted as plates and samples with low probability scores. Once the practitioners got more confident about the data, they commented on their desire to include functionality to filter out high-confidence pseudo-labels from the *Neighborhood view* as labeled neighbors increased their trust compared to unlabeled ones. Saliency maps helped validate cases when the image of interest shared patterns with the neighbors and was mislabeled (Fig. 6e), but these heatmaps did not help explain well out-of-distribution samples (Fig. 6c).

To identify the high-confidence samples to leverage during training, most practitioners remained in the *training* subset and used the *confident* tab (*T2.4*). After iterating over that panel, three practitioners spotted out-of-distribution samples for the minority class (gels). P4 preferred to inspect the unlabeled subset alone and use filters to visualize the regions for high-confidence samples (*T1.2*). He explained that the lack of filters in the *Gallery view* was a disadvantage; thus, he preferred to use the scatterplot. He added: *"Predicted gels look good, but plates are a bit weird; it looks like there are not many predicted plates in this unlabeled collection."* To inspect the low-confidence samples for the last task, they preferred to use the *uncertain* tab (*T2.3*). A couple of practitioners (P1, P2) commented that they needed a way to look back at labeled data as they forgot some data characteristics. In particular, one of them iterated back and forth between projections. While none of the practitioners had major troubles updating labels, one indicated his preference for selecting elements per *Neighborhood* rather than the sorted thumbnails in the *Gallery view*.

5.3. Feedback

BI-LAVA yielded excellent feedback from the ML practitioners participating in our evaluation. For example, P1 told us he has to deal with many videos in his daily work; he commented: *"(BI-LAVA) is so powerful to help a practitioner understand their data and develop ideas to improve the model. By the way, I like this tool so much, and if possible, I will consider using it in my research"*. In addition, practitioners agreed that the interface provided encodings and interactions helpful in speeding up labeling, identifying errors in labeled data, exploring and understanding the data in different ways, identifying imbalanced distributions, inspecting the quality of the pseudo-labels on high-confidence samples, and labeling low-confidence samples.

Although our system uses multiple views and entry points, practitioners commended BI-LAVA's ease of use. *"Fairly intuitive and easy to use"*, P3 commented. *"Good looking front-end, it's easy to use, and it has powerful functions"*, P1 added. Positive feedback was also given for these multiple views, filters, and supporting encodings: *"(P4 liked) the filtering and neighborhood configuration, as well as the view that filtered images by different metrics. Grad-CAM was useful in figuring out what the classifier looked at for the more simple images and finding the noise's source"*. Regarding potential improvements, two practitioners suggested adding flexibility to the panels, such as readjusting their size or showing them on demand.

The interaction between the *Projection* and *Neighborhood Views* was the primary component to explore, understand, and validate the data. In addition, our spatial spiral layout served as a proxy for the projected image features: *"... I was mainly interested in the neighborhood as a proxy for similar points, and a way to get more details on them, and less in terms of ranked similarity to a given image"*, commented P4. Three practitioners also stated their preference for our spatial spiral encoding, highlighting its consistency with the projected samples; one preferred any spiral variation (P2), and one stated he could use any layout (P3). Regarding labeling, practitioners liked the multi-instance selection feature; P2 commented *"having as many images as possible on the page is best"*.

Last but not least, P5 commented: *"As a junior researcher and active competitor in ML contests, I spend more time designing models than visualizing datasets. I usually do error analysis of the top mispredictions to understand why my model might fail. I usually use Python and slowly interact with Jupyter. But, while using the interface, I noticed that this process can be improved and accelerated."*

6. Discussion

Our evaluation shows that BI-LAVA aids in understanding the characteristics of different classes in a hierarchy, validating the quality of data and models, and labeling an unlabeled collection of biomedical images. Although BI-LAVA's multi-view interface is complex at first glance, our evaluation shows that ML practitioners could use it properly after only a few minutes. Furthermore, an expert audience adapted quickly to the interface complexity and provided positive feedback on usability.

Results from our case study with a domain expert highlight the importance of reviewing data quality for AL. The *Projection view* helped the user spot mislabeled images. The target exploration in the *Gallery* view led to identifying mistakes in the extraction pipeline. For all cases, the *Neighborhood* view was essential to compare images and expand the search for similar images, which supports our multi-view design. Identifying errors through our interface supported the evaluation of his extraction and modeling tools on a broader data scope.

Results from our second case study with ML practitioners demonstrate BI-LAVA's potential to familiarize users with an unfamiliar dataset, regardless of their familiarity with data visualization. Our multiple-classifier strategy helped users focus on particular data subsets while BI-LAVA leveraged incomplete ground-truth labels to provide information in the form of latent features and predictions. In addition, through multiple exploration strategies involving single and multiple views, users could explain the characteristics of each data class, identify and correct mislabeled instances and even reason about issues in parent classifiers that inserted out-of-distribution samples to the inspected subset.

The literature on visual interactive labeling formalizes several strategies for single-view exploration [BHS*21]. We confirm these strategies are also used in our *Projection* view. For instance, users explored the projected samples by examining areas with low and high density first, inspecting overlapping regions first, checking elements with disagreement among neighbors, or sampling thumbnails among areas for coverage.

While a single view can support many labeling strategies [BHS*21], our work shows that multiple connected views further expand these actions. Exploration following an overview-first approach [Shn96] benefited the practitioners with no previous experience with our dataset as it allowed them to leverage the spatial positions on the layout or data characteristics through filters to get a better grasp of the data. Once users got more familiar with the dataset, the details-first approach [LBS*18] through the gallery became more widely used. In particular, users started to use the feedback from the AL backend when inspecting low- and high-confidence samples. The preference for displaying thumbnails in spiral layouts in contrast to a grid layout suggests the benefits of maintaining the spatial similarity between items. Our spatial spiral layout matched the projected view more closely than the spiral layout. In addition, we observed that displaying thumbnails in a juxtaposed view, as opposed to a superimposed one, encouraged more comparisons between regions, as users can visualize more images simultaneously without clutter.

Our second case study also suggests that the user guidance provided by our neighborhood view is helpful when exploring projected data and avoiding misleading exploration strategies. In particular, after projecting the data using t-SNE, one user explored first groups of images further from the overlapping areas between class boundaries. Then, progressively the user looked at groups closer to the overlapping region while checking the variations among image groups. The rationale was that groups closer to an overlapping area might have similar features to another image class. However, the distortions resulting from t-SNE may provide incorrect cues for the t-SNE projection used. As Wattenberg et al. [WVJ16] mentioned, distances may not map to the user's intuition and are tightly coupled

to the perplexity parameter, which we did not allow users to manipulate. Enhancing our *Projection view* with encodings to show distortions can avoid these wrong conclusions [NA18]. We minimize these problems by using the *Neighborhood View* as a secondary layout enrichment [NA18], which requires additional interaction from the user.

Reflecting upon this experience, we identified the following three lessons for researchers addressing similar challenges:

- *L1. Use complex multiple views, as long as they support desired functionality.* Despite the common advice to use simple interfaces for beginners, our clients valued the multiple coordinated views and appreciated the system's enhanced functionality.
- *L2. Show neighborhood information to support trustworthiness.* Our clients prioritized identifying trustworthy data samples over precise similarity ranking. Maintaining neighborhood spatiality in reduced embeddings and using connected views for thumbnails were helpful, but researchers should note and communicate potential projection distortions to users.
- *L3. Provide slicing support to find and contrast trusted items.* Clients unfamiliar with the data used trustworthy samples as reference points. Comparing different data slices, such as train/validation/test sets, against unlabeled sources aided in assessing model generalizability and data quality.

Regarding limitations, BI-LAVA assumes that model builders can learn a new dataset through progressive exploration by leveraging pre-trained classifiers. However, we did not consider more complex bootstrapping scenarios. Next, due to our goal of providing an efficient labeling tool, we analyzed the functionality of multiple views and did not compare performance metrics between layouts. As BI-LAVA's front end only depends on algorithms to classify low and high-probability samples, more recent AL or other strategies could easily replace the CEAL component. Model training can be further expanded to use self-supervised learning in an AL setting [MVBA21] to improve the use of unlabeled data. Finally, GradCAM might fail to produce correct explanations and can fail to adversarial attacks [VWLE19], where appropriate. Our support for neighborhood exploration, however, helps to alleviate this issue by allowing users to explore additional evidence.

BI-LAVA is designed for scalability. Our *Neighborhood view* displays and enables labeling up to 180 images at once, and we have been able to render up to 850,000 images simultaneously in the *Projection view*. The color and shape patterns support pre-attentive similarity detection [MM18], even for images on the spiral periphery, which may be less legible. Zooming, filtering, and visual summaries help navigate the data effectively. However, the *Gallery* view's design, supported by our AL strategy, attempts to show prioritized elements to identify regions of interest with images of low and high confidence. Thus, this view does not attempt to scale to avoid a cognitive overload. Similarly, our dataset view relies on aggregated data but is limited in the number of classes shown at each node in the taxonomy.

Although we initially developed BI-LAVA to label biomedical images, it can generalize to other image classification problems in a semi-supervised setting by updating the input classification taxonomy, which can also be non hierarchical, as long as labeling samples are provided for those taxonomies. BI-LAVA's methodology could also be adapted to non-image classification problems using a suitable visual representation of the data items to replace the image thumbnails. Future work directions include further generalizability to other domains by enabling visual editing of taxonomies, supporting parametrization of the dimensionality reduction algorithms [CMK20], and providing a visual summary of the changes in model performance across iterations.

7. Conclusion

In this work, we described the design and evaluation of BI-LAVA, a novel and timely labeling system initially developed for the hierarchical labeling of image modalities in biocuration. BI-LAVA integrates a visual analytics interface and an ML strategy for deep image classifiers to help non-experts understand a biomedical dataset, correct data quality issues, label samples, and reason about model behavior. In addition, BI-LAVA uses multiple classifiers to support a hierarchical taxonomy that deals with incomplete ground-truth labels. We described the characteristics of these biomedical images and identified the requirements based on a long-term collaboration with biocurators and text-mining researchers in the biocuration domain. Furthermore, we introduced custom views that support the understanding and labeling of unfamiliar datasets and exploring neighborhoods of thumbnails leveraging spatial constraints. Finally, our evaluation with machine learning practitioners and collaborators proves the usefulness of BI-LAVA, with a particular interest in the familiarization of non-experts with an image dataset.

Conflict of Interest

None declared.

Data Availability

The data used to train the image classifiers of this study are available from the corresponding author upon reasonable request. Trained image models and source code are available at https://github.com/uic-evl/bio-search.

Supplementary Materials

Video demonstration, additional layouts, and spiral layout code are available at https://osf.io/nvbtp/?view_only=d9aea97c77944916abadd8e65f9eee2f

Acknowledgements

References

[AA20] ABUTHAWABEH A., AUPETIT M.: A Force-Directed Power Diagram Approach for Interactive Voronoi Treemaps. In *Proc. EuroVis - Short Papers* (2020), The Eurographics Association, pp. 109–113. 3

[Ado19] ADOBE RESEARCH: CHART-Synthetic. https://github.com/adobe-research/CHART-Synthetic, 2019. 4, 5

[Ana22] ANACONDA: *State of Data Science Report.* Tech. rep., Anaconda Inc, 2022. 2

[Bai21] BAIDAKOVA D.: Data Labelers in Highly Specialized Fields: How and Where Do We Get Them? https://bit.ly/baidakova-medium-label, 2021. 1

[BHA*22] BERTUCCI D., HAMID M. M., ANAND Y., RUANGROTSAKUN A., TABATABAI D., PEREZ M., KAHNG M.: Dendromap: Visual exploration of large-scale image datasets for machine learning with treemaps. *IEEE TVCG* (2022), 1–11. 3

[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 TiiS 11*, 3–4 (2021), 1–42. 2, 4, 12

[BHZ*18] BERNARD J., HUTTER M., ZEPPELZAUER M., FELLNER D., SEDLMAIR M.: Comparing Visual-Interactive Labeling with Active Learning: An Experimental Study. *IEEE TVCG 24*, 1 (2018), 298–308. 2

[BLP19] BURNS G. A., LI X., PENG N.: Building deep learning models for evidence classification from the open access biomedical literature. *Database* (2019). 3

[BNR20] BÄUERLE A., NEUMANN H., ROPINSKI T.: Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks. *CGF 39*, 3 (2020), 195–205. 2

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D³ data-driven documents. *IEEE TVCG 17*, 12 (2011), 2301–2309. 6

[BZSA18] BERNARD J., ZEPPELZAUER M., SEDLMAIR M., AIGNER W.: VIAL: a unified process for visual interactive labeling. *Visual Computer 34*, 9 (2018), 1189–1207. 1, 2, 3, 4

[CMC*21] CUTURA R., MORARIU C., CHENG Z., WANG Y., WEISKOPF D., SEDLMAIR M.: Hagrid—gridify scatterplots with hilbert and gosper curves. In *Proc. VINCI* (2021), pp. 1–8. 3

[CMK*20] CHATZIMPARMPAS A., MARTINS R. M., KERREN A.: t-visne: Interactive assessment and interpretation of t-SNE projections. *IEEE TVCG 26*, 8 (2020), 2696–2714. 13

[CSV*18] CHEN N.-C., SUH J., VERWEY J., RAMOS G., DRUCKER S., SIMARD P.: Anchorviz: Facilitating classifier error discovery through interactive semantic data exploration. *ACM TiiS* (2018), 269–280. 2, 8

[CWM03] COHEN W. W., WANG R., MURPHY R. F.: Understanding captions in biomedical publications. In *Proc. ACM SIGKDD* (2003), pp. 499–504. 3

[CWW*20] CHEN C., WANG Z., WU J., WANG X., ET AL.: Interactive Graph Construction for Graph-Based Semi-Supervised Learning. *IEEE TVCG 27*, 9 (2020), 3701–3716. 2, 3

[CYL*21] CHEN C., YUAN J., LU Y., LIU Y., SU H., YUAN S., LIU S.: OoDAnalyzer: Interactive Analysis of Out-of-Distribution Samples. *IEEE TVCG 27*, 7 (2021), 3335–3349. 3

[DFAST12] DEMNER-FUSHMAN D., ANTANI S., SIMPSON M., THOMA G. R.: Design and Development of a Multimodal Biomedical Information Retrieval System. *J. Comput. Sci. Eng. 6*, 2 (2012), 168–177. 4, 5

[DGK*10] DOW S. P., GLASSCO A., KASS J., SCHWARZ M., SCHWARTZ D. L., KLEMMER S. R.: Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM TOCHI 17*, 4 (2010), 1–24. 6

[DLSP18] DUDÁŠ M., LOHMANN S., SVÁTEK V., PAVLOV D.: Ontology visualization methods and tools: a survey of the state of the art. *Knowl. Eng. Rev. 33* (2018). 3

[DSF*14] DUARTE F. S., SIKANSI F., FATORE F. M., FADEL S. G., PAULOVICH F. V.: Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE TVCG 20*, 12 (2014), 2063–2071. 3

[FDH*15] FRIED O., DIVERDI S., HALBER M., SIZIKOVA E., FINKELSTEIN A.: Isomatch: Creating informative grid layouts. In *CGF* (2015), vol. 34, pp. 155–166. 3

[GCB*15] GHONIEM M., CORNIL M., BROEKSEMA B., STEFAS M., OTJACQUES B.: Weighted maps: treemap visualization of geolocated quantitative data. In *Vis. Data Anal.* (2015), vol. 9397, pp. 163–177. 3

[GHM*22] GÖRTLER J., HOHMAN F., MORITZ D., WONGSUPHASAWAT K., REN D., NAIR R., KIRCHNER M., PATEL K.: Neo: Generalizing confusion matrix visualization to hierarchical and multi-output labels. In *Proc. CHI* (2022). 3

[GNCM*16] GOMEZ-NIETO E., CASACA W., MOTTA D., HARTMANN I., TAUBIN G., NONATO L. G.: Dealing with multiple requirements in geometric arrangements. *IEEE TVCG 22*, 3 (2016), 1223–1235. 3

[Gra23] GRAND VIEW RESEARCH: *Data Collection And Labeling Market Size, Share & Trends Analysis Report By Data Type (Audio, Image/ Video, Text), By Vertical (IT, Automotive, Government, Healthcare, BFSI), By Region, And Segment Forecasts, 2023 - 2030.* Tech. Rep. GVR-4-68038-406-2, Grand View Research Inc, March 2023. 1

[GSdHSBM16] GARCIA SECO DE HERRERA A., SCHAER R., BROMURI S., MULLER H.: Overview of the ImageCLEF 2016 medical task. In *Working Notes CLEF* (2016). 3, 4

[GZL*20] GOU L., ZOU L., LI N., HOFMANN M., SHEKAR A. K., WENDT A., REN L.: VATLD: A visual analytics system to assess, understand and improve traffic light detection. *IEEE TVCG 27*, 2 (2020), 261–271. 1

[HNH*12] HÖFERLIN B., NETZEL R., HÖFERLIN M., WEISKOPF D., HEIDEMANN G.: Inter-Active Learning of Ad-Hoc Classifiers for Video Visual Analytics. In *Proc. VAST* (2012), pp. 23–32. 2

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Proc. CVPR* (2016), pp. 770–778. 5

[Int18] INTERNATIONAL SOCIETY FOR BIOCURATION: Biocuration: Distilling data into knowledge. *PLoS Biology 16*, 4 (2018), 1–8. 3

[JLK*20] JIANG X., LI P., KADIN J., BLAKE J. A., RINGWALD M., SHATKAY H.: Integrating image caption information into biomedical document classification in support of biocuration. *Database* (2020). 3

[KKZE19] KHAYAT M., KARIMZADEH M., ZHAO J., EBERT D. S.: VASSL: A visual analytics toolkit for social spambot labeling. *IEEE TVCG 26*, 1 (2019), 874–883. 1, 2

[KL83] KRUSKAL J. B., LANDWEHR J. M.: Icicle plots: Better displays for hierarchical clustering. *Am. Stat. 37*, 2 (1983), 162–168. 3

[LBS*18] LUCIANI T., BURKS A., SUGIYAMA C., KOMPERDA J., MARAI G. E.: Details-first, show context, overview last: supporting exploration of viscous fingers in large-scale ensemble simulations. *IEEE TVCG 25*, 1 (2018), 1225–1235. 8, 12

[LCL*19] LIU S., CHEN C., LU Y., OUYANG F., WANG B.: An Interactive Method to Improve Crowdsourced Annotations. *IEEE TVCG 25*, 1 (2019), 235–245. 2, 3

[LJKS18] LI P., JIANG X., KAMBHAMETTU C., SHATKAY H.: Compound image segmentation of published biomedical figures. *Bioinformatics* (2018), 1192–1199. 5

[LJS19] LI P., JIANG X., SHATKAY H.: Figure and caption extraction from biomedical documents. *Bioinformatics* (2019), 4381–4388. 5

[LJZ*21] LI P., JIANG X., ZHANG G., TRELLES J., RACITI D., SMITH C., RINGWALD M., MARAI G. E., ARIGHI C., SHATKAY H.: Utilizing image and caption information for biomedical document classification. *Bioinformatics 37* (2021), i468–i476. 3

[LZC*20] LEKSCHAS F., ZHOU X., CHEN W., GEHLENBORG N., BACH B., PFISTER H.: A generic framework and library for exploration of small multiples through interactive piling. *IEEE TVCG 27*, 2 (2020), 358–368. 3

[Mar17] MARAI G. E.: Activity-centered domain characterization for problem-driven scientific visualization. *IEEE TVCG* (2017), 913–922. 4

[MBS*10] MOEHRMANN J., BERNSTEIN S., SCHEGEL T., WERNER G., HEIDERMANN G.: Improving the Usability of Hierarchical Representations for Interactively Labeling Large Image Data Sets. In *Proc. HCI* (2010), pp. 618–627. 2

[MG13] MAYORGA A., GLEICHER M.: Splatterplots: Overcoming Overdraw in Scatter Plots. *IEEE TVCG 19*, 9 (2013), 1526–1538. 3, 7

[MHM20] MCINNES L., HEALY J., MELVILLE J.: Umap: Uniform manifold approximation and projection for dimension reduction, 2020. 7

[MM18] MARAI G. E., MA C. E. A.: Precision risk analysis of cancer therapy with interactive nomograms and survival plots. *IEEE Trans. Vis. Comp. Graphics 25*, 4 (2018), 1732–1745. 13

[MVBA21] MARGATINA K., VERNIKOS G., BARRAULT L., ALETRAS N.: Active learning by acquiring contrastive examples. In *Proc. Conf. Empirical Met. NLP* (2021), Assoc. Comput. Linguis., pp. 650–663. 13

[NA18] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE TVCG 25*, 8 (2018), 2650–2673. 7, 13

[nap19] NAPARI CONTRIBUTORS: napari: a multi-dimensional image viewer for python. zenodo.org/record/7276432, 2019. 2

[NW08] NGUYEN G., WORRING M.: Interactive access to large image collections using similarity-based visualization. *Visual Lang. Comput. 19*, 2 (2008), 203–224. 3

[PNML08] PAULOVICH F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE TVCG 14*, 3 (2008), 564–575. 7

[RAP23] RAPIDS DEVELOPMENT TEAM: *RAPIDS: Libraries for End to End GPU Data Science*, 2023. 6

[RFT18] RAUBER P. E., FALCÃO A. X., TELEA A. C.: Projections as visual aids for classification system design. *Inf. Visualization 17*, 4 (2018), 282–305. 2, 7

[SCB06] SHATKAY H., CHEN N., BLOSTEIN D.: Integrating image data into biomedical text categorization. *Bioinformatics 22*, 14 (2006), e446–e453. 3

[SCD*17] SELVARAJU R. R., COGSWELL M., DAS A., VEDANTAM R., PARIKH D., BATRA D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE ICCV* (2017), pp. 618–626. 9

[SDW01] SCHEFFER T., DECOMAIN C., WROBEL S.: Active hidden markov models for information extraction. In *Proc. IDA* (2001), pp. 309–318. 4, 5

[Set09] SETTLES B.: *Active Learning Literature Survey*. Tech. rep., Univ. Wisconsin-Madison Department Computer Sciences, 2009. 1

[SG17] SARIKAYA A., GLEICHER M.: Scatterplots: Tasks, data, and designs. *IEEE TVCG 24*, 1 (2017), 402–412. 4

[Sha48] SHANNON C. E.: A mathematical theory of communication. *The Bell System Tech. J. 27*, 3 (1948), 379–423. 4, 5

[Shn92] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graphics 11*, 1 (jan 1992), 92–99. 3

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. *ACM Trans. Graphics* (1996), 336–343. 7, 12

[SJZ21] SAGER C., JANIESCH C., ZSCHECH P.: A survey of image labelling for computer vision applications. *Business Analytics 4*, 2 (2021), 1–20. 1

[SS18] SENER O., SAVARESE S.: Active learning for convolutional neural networks: A core-set approach. In *Proc. ICLR* (2018). 2

[SSKEA19] SPERRLE F., SEVASTJANOVA R., KEHLBECK R., EL-ASSADY M.: Viana: Visual interactive annotation of argumentation. In *Proc. VAST* (2019), IEEE, pp. 11–22. 2

[SSZ*17] SACHA D., SEDLMAIR M., ZHANG L., LEE J. A., PELTONEN J., WEISKOPF D., NORTH S. C., KEIM D. A.: What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing 268* (2017), 164–175. 2

[TASM23] TRELLES J., ARIGHI C., SHATKAY H., MARAI G. E.: Enhancing biomedical search interfaces with images. *Bioinformatics Advances* (07 2023), vbad095. 1, 3

[The23] THE BATCH: The secret life of data labelers. https://www.deeplearning.ai/the-batch/issue-204/, July 2023. 1

[TLA*20] TRELLES J., LI P., ARIGHI C., SHATKAY H., MARAI G. E.: Modality-classification of microscopy images using shallow variants of deep networks. In *IEEE Int. Conf. Bioinf. Biomed. (BIBM)* (2020), IEEE, pp. 2379–2385. 1

[TLA*21] TRELLES J., LI P., ARIGHI C., RACITI D., SHATKAY H., MARAI G. E.: ANIMO: Annotation of Biomed Image Modalities. In *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)* (2021), pp. 1069–1076. 1, 3, 4, 5, 9

[TMHL20] TKACHENKO M., MALYUK M., HOLMANYUK A., LIUBIMOV N.: Label Studio: Data labeling software. github.com/heartexlabs/label-studio, 2020. 2

[VBW15] VEHLOW C., BECK F., WEISKOPF D.: The state of the art in visualizing group structures in graphs. In *Proc. EuroVis - STARs* (2015), pp. 21–40. 3

[vdWKB20] VAN DE WETERING H., KLAASSEN N., BURCH M.: Space-reclaiming icicle plots. In *Proc. PacificVis* (2020), pp. 121–130. 3

[vH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *J. Mach. Learn. Res. 9*, 11 (2008). 7

[VWLE19] VIERING T., WANG Z., LOOG M., EISEMANN E.: How to manipulate CNNs to make them lie: the GradCAM case, 2019. 13

[WLC*20] WANG L. L., LO K., CHANDRASEKHAR Y., REAS R., YANG J., ET AL.: CORD-19: The COVID-19 Open Research Dataset. In *Proc. Workshop NLP COVID-19 ACL* (2020). 4, 5

[WRZ*15] WANG C., REESE J. P., ZHANG H., TAO J., GU Y., MA J., NEMIROFF R. J.: Similarity-based visualization of large image collections. *Inf. Visualization 14*, 3 (2015), 183–203. 3, 8

[WVJ16] WATTENBERG M., VIÉGAS F., JOHNSON I.: How to Use t-SNE Effectively. *Distill* (2016). 12

[WZL*17] WANG K., ZHANG D., LI Y., ZHANG R., LIN L.: Cost-effective active learning for deep image classification. *IEEE Trans. Circuits Syst. Video Tech. 27*, 12 (2017), 2591–2600. 2, 5

[XXX*19] XIANG S., XI Y., XIA J., WU J., CHEN Y., LIU S.: Interactive Correction of Mislabeled Training Data. In *Proc. VAST* (2019), pp. 57–68. 2, 3

[YAJC09] YU H., AGARWAL S., JOHNSTON M., COHEN A.: Are figure legends sufficient? Evaluating the contribution of associated text to biomedical figure comprehension. *J. Biomed. Discovery Collab. 4*, 1 (2009). 3

[YCY*20] YUAN J., CHEN C., YANG W., LIU M., XIA J., LIU S.: A survey of visual analytics techniques for machine learning. *Comput. Visual Media* (2020), 1–34. 2