Department of Computer Science
Texas State University

ASSISTED LABELING VISUALIZER (ALVI): ADVANCING A SEMI-AUTOMATIC LABELING SYSTEM FOR TIME-SERIES DATA

Tristan Rech
jjp176@txstate.edu
CS4395 – Independent Study Report – Spring 2024

Abstract

Machine learning applications significantly benefit from large amounts of accurately labeled data, yet the task of labeling remains notoriously challenging and time-consuming. This is particularly evident in domains involving human subjects, where labeling time-series signals often necessitates trained professionals. Building upon the original Assisted Labeling Visualizer (ALVI), which simplified the labeling process through an interactive interface visualizing synchronized video, feature-map representations, and raw time-series signals, our work introduces a major overhaul of the user interface and enhances the overall functionality of the system. We have developed a centralized interface that integrates these technologies, making the system more intuitive and user-friendly while maintaining its robust deep learning and self-supervised learning frameworks. These improvements make ALVI more functional and efficient across all interfaces, significantly streamlining the semi-automatic labeling of large volumes of unlabeled data. We demonstrate the enhanced system's capabilities on a human activity recognition dataset, showcasing its expanded potential to facilitate diverse machine learning applications by simplifying the time-series data labeling process.

1 INTRODUCTION

Data labeling is a crucial step in the process of machine learning applications. It involves assigning relevant and accurate tags to data that are used to train models. Labeling time-series data is particularly important for applications that involve continuous monitoring and tracking of data, such as in healthcare, manufacturing, and environmental monitoring. Time-series sensor data can provide valuable insights into changes in a system, environment, or individual's behavior over time. By accurately labeling such data, machine learning models can identify patterns and predict future outcomes.

The utilization of time-series sensor data and labeling can significantly enhance accessibility for people with disabilities. For instance, by deploying sensors throughout a museum and labeling the captured data, machine learning models can be trained to recognize patterns in visitor behavior [13], such as popular exhibits and frequently taken routes. This information can then be utilized to provide more accessible experiences for people with disabilities [20][10].

Manual labeling of time-series sensor data can be a daunting and challenging task, particularly when dealing with massive and complex datasets. One of the main difficulties is that humans are not naturally skilled at reading and interpreting raw time-series sensor data. Even with the use of video data to assist in the labeling process, the manual labeling of time-series data can still be tedious, time-consuming, and error-prone. The process of manual labeling requires significant human resources and can lead to inconsistencies across different human labelers. Semi-automatic labeling can make use of machine learning algorithms to pre-label the data and allow human labelers to correct any errors or inconsistencies in the pre-labeled data. This approach can significantly reduce the time and effort required for manual labeling and improve the accuracy and consistency of the labeled data.

In this work, we introduce a browser-based software framework[1] for labeling time-series sensor data that incorporates state-of-the-art visualization and machine learning techniques, enabling efficient and precise semi-automatic labeling of such data. We employ interactive visualizations of raw time-series data, as well as features extracted through semi-supervised learning methods. Furthermore, we incorporate a label correction process to detect and correct any potential errors in the automatically assigned labels. The human remains in the loop to ensure the quality of the assigned labels but with a significantly reduced workload. Our system is compatible with any Python-capable machine, thus enabling local execution to maintain data confidentiality in cases where the data is sensitive.

2 RELATED WORK

There have been many general systems developed to label data, such as text, images, and audio. One example is the VGG Image Annotator [8]. Crowd-sourcing platforms like Amazon Mechanical Turk [3], Apen [4], and Labelbox [15] are commonly used for general data labeling. These platforms provide a cost-effective and scalable solution for data labeling, but they have certain drawbacks, such as the difficulty in ensuring the quality of labels and the potential for low-quality labels due to the lack of expertise of the workers. General data labeling systems can have issues with ambiguity, making it difficult for labelers to accurately label data.

Labeling time-series sensor data presents a unique set of challenges when compared to labeling images, videos, or text. Examples of labeling systems used for time-series sensor data include Visplore [2] and Label Studio [1]. These systems use various techniques such as manual labeling, rule-based labeling, and semi-automatic labeling to overcome these challenges. However, the labeling of time-series data is often more tedious and time-consuming than other types of data.

Semi-automatic labeling is a method that combines human expertise with machine learning algorithms to improve the efficiency and accuracy of data labeling. Examples of literature that use semi-automatic labeling include VAST [7] and SALT [19]. Semi-automatic labeling has been shown to significantly reduce the time and effort required for manual labeling while maintaining high labeling accuracy. However, most available solutions so far only apply to image/video and textual data.

Data visualization tools, such as t-SNE [21] and UMAP [17], can assist in the labeling of time-series sensor data by providing visual representations of the data that can aid in identifying patterns and relationships. UMAP has been shown to be particularly useful for dimensionality reduction and visualization of high-dimensional time-series sensor data.

There are several techniques available for automatically correcting mislabeled instances of time series data. These methods generally employ deep learning systems that can recognize the correct class of mislabeled instances, despite the difficulty that machine learning models may encounter when training on noisy labels [5]. By comparing the output of a trained convolutional neural network (CNN) to the assigned labels in a dataset, it is possible to identify which instances in a sensor dataset are most likely to be mislabeled [6]. Another approach for identifying mislabeled data is to compare instances to their nearest neighbors and determine the most probable correct label either by comparison to neighbors [22] or statistical inference [23].

3 METHODOLOGY

The labeling framework introduced in this study has been transformed into a versatile Dash application, serving as a central interface for various labeling tasks. This application, now available as a standalone package, can be easily installed via pip and seamlessly integrated into any Python capable environment. Leveraging the power of Plotly Dash, the web-based tool retains its compatibility with all major browsers while offering enhanced functionality beyond the confines of Jupyter Notebooks [18]. By

[1] `https://github.com/imics-lab/time-series-label-assist`

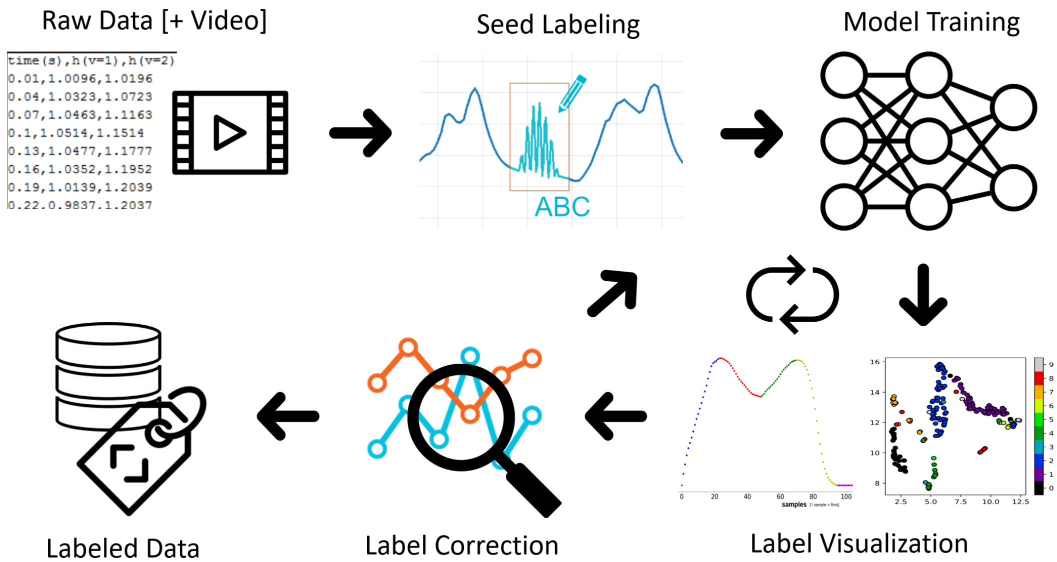


Figure 1: ALVI semi-automatic labeling process workflow.

providing a self-contained solution, our framework ensures secure and privacy-preserving labeling of sensitive data, eliminating the need for data upload to remote servers for annotation.

The data labeling process comprises the following steps:

1. Raw time series data, and if available video, are loaded from a file.
2. Data and corresponding video are visualized as time-series plots using Plotly.
3. A small amount of labeled data are provided or manually labeled by the user.
4. A deep learning model is trained based on the initially labeled data.
5. A larger amount of data is automatically labeled using the trained model.
6. The automatically assigned labels are analyzed, and parts of the data are manually reviewed.

Figure 1 visually depicts the above workflow. In the following subsections, we elaborate on the methods and tools used in each one of the steps above.

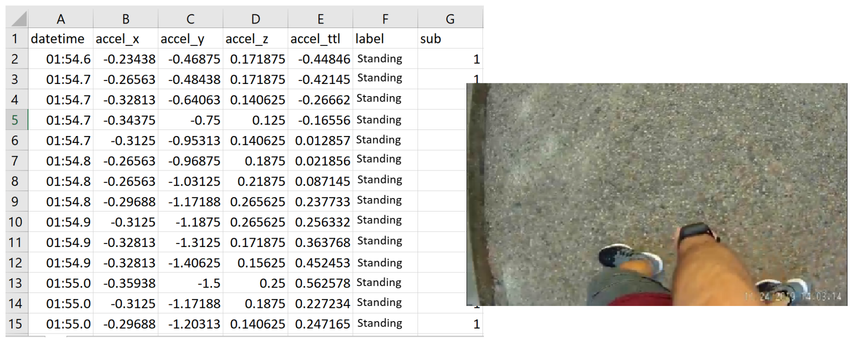
3.1 Data Loading

The current version of the software supports data files in CSV text format. It is assumed that files are structured as: <timestamp>,<ch1>,...,<chn>,<label>,<sub> where `timestamp` is the time stamp of the sensor measurement, `ch1,...,chn` represent the different data channels. For example, an accelerometer may have three channels for the three axes of acceleration, XYZ. The `label` column represents a label assigned to each time step of data. Initially, that value can be set to `‘‘other’’` if the data is unlabeled. The `sub` column represents the subject from which the data were collected and can be a number or a string. When data from multiple subjects are aggregated together, it is often necessary to maintain subject independence in machine learning experiments. Thus, maintaining the subject label is desirable. Each row of the file is a sensor measurement.

Along with the raw time-series data, the user can specify a video file, if available, to inform the labeling process (e.g. Fig. 2a). The loaded video file can be synchronized with a particular section of the raw data by specifying an offset with respect to the data timestamps.

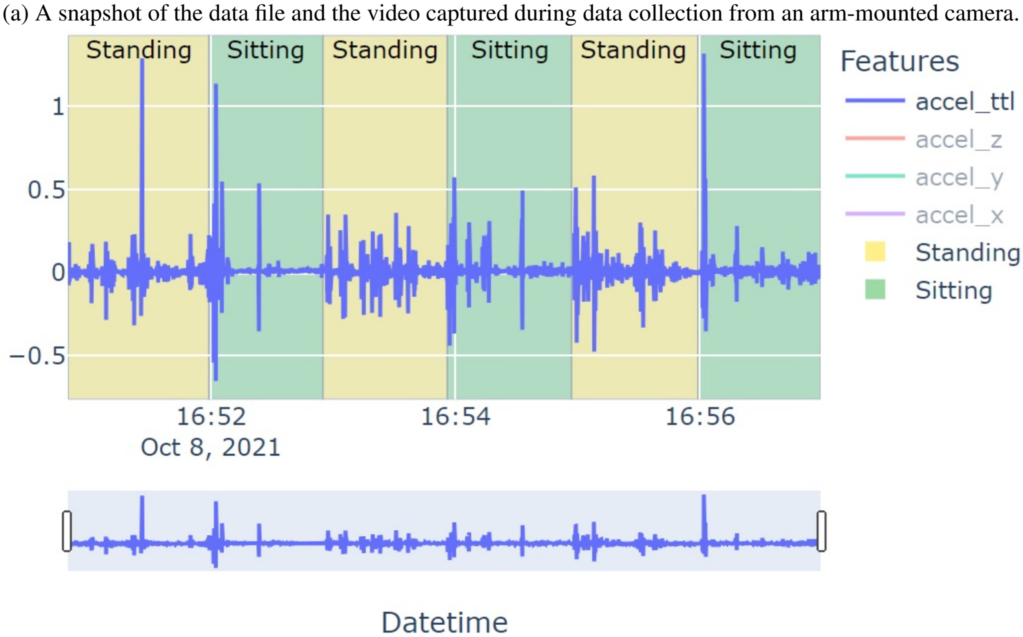
To enhance the user experience in data preparation, the software now provides a comprehensive interface for all necessary preprocessing steps. This interface allows users to adapt their data to the required format, clean it by addressing null values, and apply basic principles ensuring the data's usability. Furthermore, the software supports various configurations for further processing, all added to a configuration file. Users can specify an offset for synchronizing video files with data timestamps, crucial for aligning video and data seamlessly. Additionally, the software offers options to define the window and step size when transforming the data into windows, a method used for segmenting the data for analysis. Other configurations are also available, allowing users to tailor the preprocessing to their specific needs and objectives. This preprocessing interface complements the initial data loading capabilities, ensuring a smooth and efficient setup for data analysis.

3.2 Seed Labeling



	A	B	C	D	E	F	G
1	datetime	accel_x	accel_y	accel_z	accel_ttl	label	sub
2	01:54.6	-0.23438	-0.46875	0.171875	-0.44846	Standing	1
3	01:54.7	-0.26563	-0.48438	0.171875	-0.42145	Standing	1
4	01:54.7	-0.32813	-0.64063	0.140625	-0.26662	Standing	
5	01:54.7	-0.34375	-0.75	0.125	-0.16556	Standing	
6	01:54.7	-0.3125	-0.95313	0.140625	0.012857	Standing	
7	01:54.8	-0.26563	-0.96875	0.1875	0.021856	Standing	
8	01:54.8	-0.26563	-1.03125	0.21875	0.087145	Standing	
9	01:54.8	-0.29688	-1.17188	0.265625	0.237733	Standing	
10	01:54.9	-0.3125	-1.1875	0.265625	0.256332	Standing	
11	01:54.9	-0.32813	-1.3125	0.171875	0.363768	Standing	
12	01:54.9	-0.32813	-1.40625	0.15625	0.452453	Standing	
13	01:55.0	-0.35938	-1.5	0.25	0.562578	Standing	
14	01:55.0	-0.3125	-1.17188	0.1875	0.227234	Standing	
15	01:55.0	-0.29688	-1.20313	0.140625	0.247165	Standing	1

(a) A snapshot of the data file and the video captured during data collection from an arm-mounted camera.



(b) An example visualization used for seed labeling. The raw signals, color-coded assigned label line, and confidence lines are visible.

Figure 2: Data loading and seed labeling stages.

In order to facilitate the automatic labeling of the remaining data, it is necessary to have a small amount of initial labeled data. These initial labeled data can either be directly loaded into the system if they have already been pre-labeled outside of the system or manually labeled within the system by utilizing the visualization capabilities and the video-sync feature.

The seed labeling process can use advanced visualization features and interactive graphs provided by Plotly. For instance, the user is afforded the ability to zoom in on the signal, select specific sections, and assign labels by specifying the starting and ending points of a segment, the label, and their confidence level regarding the assigned label (i.e., low, medium, or high). Confidence levels can be utilized in model training. For example, segments of low confidence can be excluded from the training set.

To further refine the seed labeling process and reduce the potential for error, significant improvements have been made to the user interface (see Fig. 2b). Previously, the interface required manual entry of start and end times using the datetime index format (YYYY-MM-DD HH:MM:SS), which was prone to errors due to the complexity of manually typing these timestamps. To streamline this process and enhance accuracy, new functionality has been introduced that allows users to select a time window directly on the graph. By clicking and dragging across the desired range, users can automatically fill in the start and end times. This feature leverages Plotly's existing zoom capabilities to capture the x0 and xn coordinates—where n denotes the size of the window—and accurately assigns the corresponding datetime values to these points. This adaptation not only simplifies the user input process but also significantly reduces the likelihood of errors associated with manual data entry, ensuring more reliable and efficient data labeling within the system.

In addition to the enhancements in data entry, several improvements have been implemented to optimize the graph visualization for labeling. One significant change is the removal of the label line, replaced by labels that appear directly behind the corresponding time series data. These text labels are designed to display only when they do not overlap with others, ensuring clarity and readability. Additionally, the graph now supports the capability to display all available features at once, while initially only showing those selected by the user, allowing for a cleaner and more focused initial view.

A custom tooltip feature has also been introduced, which provides detailed information when users hover over any point on the graph. This interactive tooltip displays the datetime value, sensor reading, assigned label, and the confidence level of that label, giving users immediate access to detailed contextual information. This feature not only enhances the interactivity of the graph but also aids in more accurate and informed labeling, making the entire process more intuitive and efficient.

3.3 Model Training and Automatic Labeling

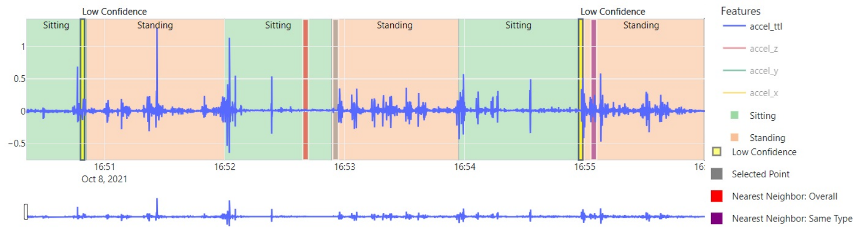
The process of training a deep learning model involves utilizing the manually pre-labeled portion of the data. During this stage, the continuous signal is segmented into fixed-size windows, and each window is assigned a single label. This technique is a commonly employed approach for training models on time-series data. The training set can comprise either overlapping or non-overlapping segments. In the event that a window spans two or more labels, the user can elect to assign the majority of time steps as the primary label for that segment or exclude the segment entirely from the training set.

The deep learning model can be based on any neural network architecture that supports time-series classification. The current version of the software primarily utilizes a convolutional neural network (CNN) for time-series classification. However, it also accommodates flexibility in model selection. Users can integrate alternative neural network architectures, such as transformer-based or long short-term memory (LSTM) models, depending on their specific needs and the availability of pretrained models. It is important to note, though, that while the software supports the integration of these different architectures, it does not currently support retraining such models with newly labeled data. This limitation is a consideration for users who might require ongoing model refinement based on fresh labels.

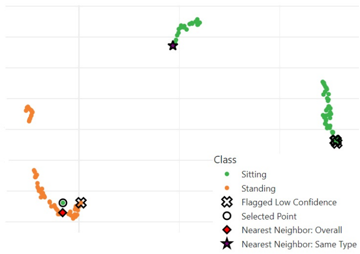
Upon acquiring a trained model, a larger quantity of data can be automatically labeled by having the model predict the correct label for each segment of the new data. Although this process saves time and effort from manual labor, it is anticipated to produce some incorrect predictions. To rectify these inaccuracies and establish a reliable ground truth, we utilize a complex approach for incorrect label detection and correction, which is detailed in the following subsections. This approach requires human intervention; however, it is substantially less labor-intensive than fully manual labeling.

3.4 Data Visualization

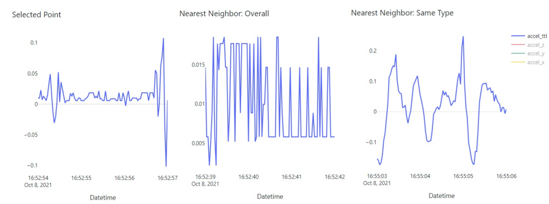
Following the training of the model and the initial auto-labeling stage, the focus shifts to data visualization to refine and verify the accuracy of the labels. Incorporating a human in the loop during the label correction process is crucial, necessitating informative visualizations to aid in decision-making. Our proposed solution involves the use of automatically labeled data that is visualized and color-coded. We employ two types of visualizations, raw signals and Uniform Manifold Approximation and Projection (UMAP) plots [17]. In the UMAP plot, each point corresponds to a signal segment. Multichannel raw signal segments are mapped to a low dimensional vector through a



(a) Visualization of the raw times series data.



(b) UMAP plot.



(c) Nearest neighbors.

Figure 3: The data visualization process utilized for label review and correction.

model trained using self-supervised contrastive learning, as presented in [9]. Such methods do not rely on the original data labels and, thus, are not affected by label noise. The low-dimensional vector is then further reduced to two dimensions for 2D visualization by UMAP.

By clicking on a point in the UMAP plot, the point is highlighted, along with the closest neighbor, irrespective of its class, and the closest neighbor of the same class as the one assigned to the point. Additionally, the corresponding signal segment in the raw time series data is also highlighted. If available, the synchronized video can display the corresponding position. Figure 3 shows an example of this process. This process can help the users decide if the automatically assigned label is correct or not. These visualizations aid the user in determining the accuracy of the automatically assigned label and in correcting it if necessary.

To enhance this interface, we initially enabled the functionality to click a UMAP point and see its representation on the raw time series, as well as its nearest neighbors. Building on this, we have now implemented the ability to interactively click on the time series plot and view the corresponding data point on the UMAP plot, along with its necessary neighbors. This bidirectional interaction is essential for comprehensively understanding the different data representations to ensure the most accurate ground truth labeling.

Additionally, we have integrated video synchronization into this functionality. Now, wherever a user clicks—whether on the UMAP or time series plot—they will jump to the beginning of that window in the video. This approach simplifies the problem to an image-type labeling process, offering an enhanced representation of what is happening in the data, assuming the video is properly aligned. The combination of these visualizations significantly enhances the effectiveness of the label correction process, ultimately leading to higher-quality labeled datasets.

3.5 Label Correction

The Assisted Labeling Visualizer (ALVI) employs advanced label correction techniques to efficiently manage the significant costs associated with manual human labeling and re-labeling of time-series data. By directing human reviewers to sections of data that are most likely mislabeled by the automatic labeling process, ALVI effectively reduces the costs and improves the efficiency of processing both new and existing datasets. This focus enhances the overall quality of the published datasets.

ALVI incorporates a label-cleaning process that leverages model prediction confidence probabilities. The system allows users to set a confidence threshold within the interface; any labels falling below this threshold are automatically flagged for review. In our visualizations, this process is distinctly marked: on the UMAP, low-confidence labels are flagged with a special X shape labeled "Flagged

Low Confidence." This marker is off by default in the legend but can be toggled on to view all the low-confidence labels. Correspondingly, in the raw time series data, these points are flagged with a yellow rectangle across the graph, also labeled "Low Confidence." However, there are no current visual indicators for low-confidence windows on the synchronized video.

These visual flags are designed to remain on the UMAP and time series graphs (unless hidden by the user) until a reviewer interacts with them. When a user selects a flagged label on either the UMAP or the raw time series plot and either verifies or changes the label, the system automatically updates the status of this data point. Upon this validation, the label is considered to exceed the confidence threshold, and the user can then proceed to the next point in need of review. This interactive approach not only highlights the areas needing attention but also ensures that corrections are efficiently integrated, streamlining the workflow and enhancing the effectiveness of data curation. This integration of detailed, interactive visual flags significantly aids reviewers in prioritizing their tasks and ensures a more informed and efficient data analysis and label correction process.

4 RESULTS

In order to evaluate the tool, we used TWristAR [12], a human activity recognition (HAR) dataset that contains multimodal data collected with an Empatica E4 Wristband. Three subjects performed scripted activities which were structured for easier labeling and balanced classes. For this work, the scripted activities were treated as labeled. Two subjects performed unscripted free-form walks that included a period of sitting as well as walking on flat ground and up/downstairs. A full video record is included. [11] provides additional dataset details and describes prior manual labeling work.

Additionally, the tool was tested on two other significant datasets to further validate its robustness and applicability across different modalities and scenarios. The first is the UCI Gesture Phase Segmentation dataset, which includes multimodal signals aimed at segmenting gestures into meaningful phases using sensors placed on the user's arms [16]. This dataset's complex gesture dynamics provided a rigorous testing environment for our tool's capabilities in handling nuanced temporal segmentation and labeling challenges.

The second dataset employed was the CMU Multimodal Activity (CMU-MMAC) Database, which collects data from subjects performing various activities in a kitchen setting, equipped with multiple sensor modalities including motion capture, video, and environmental sensors [14]. This dataset allowed us to test the efficacy of our tool in a highly unstructured environment, where activities range broadly in type and complexity.

These additional tests demonstrate the tool's versatility and efficiency in processing and labeling complex, multimodal datasets across diverse environments.

5 CONCLUSION

In this work, we introduced ALVI, a browser-based software framework that significantly advances the semi-automatic labeling of time-series sensor data. Building upon the original Assisted Labeling Visualizer, this version integrates a major overhaul in both user interface and functionality. We have enhanced the interactivity of the system through synchronized video and real-time data visualization, employing techniques like UMAP and self-supervised learning for improved data representation and labeling accuracy.

Our results demonstrate substantial improvements in labeling time reduction and accuracy across diverse datasets, including TWristAR, UCI Gesture Phase Segmentation, and the CMU Multimodal Activity Database. These tests confirm the tool's versatility and robustness in handling complex, multimodal datasets within various environments, from structured laboratory settings to dynamic, real-world scenarios.

Future work will focus on broadening the scope of ALVI to encompass additional domains that require sophisticated time-series data analysis. We aim to enrich the tool with a wider array of time-series data compatible models, possibly including LSTM or Transformer architectures. Furthermore, we plan to continuously refine the user interface and style of the application to enhance user satisfaction by reducing cognitive load and streamlining interaction. These advancements will support even more complex datasets and further automate the labeling process, ensuring both high accuracy and reduced human oversight.

References

[1] Label Studio. https://labelstud.io/, 2023. [Online; accessed 8-March-2023].

[2] Visplore. https://visplore.com/, 2023. [Online; accessed 8-March-2023].

[3] Amazon. Amazon Mechanical Turk (MTurk). https://www.mturk.com/, 2023. [Online; accessed 8-March-2023].

[4] Apen. Apen. https://appen.com/platform-5/, 2023. [Online; accessed 8-March-2023].

[5] Gentry Atkinson and Vangelis Metsis. A survey of methods for detection and correction of noisy labels in time series data. In *Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 International Conference, AIAI 2021, Hersonissos, Crete, Greece, June 25–27, 2021, Proceedings 17*, pages 479–493. Springer, 2021.

[6] Gentry Atkinson and Vangelis Metsis. Tsar: a time series assisted relabeling tool for reducing label noise. In *The 14th PErvasive Technologies Related to Assistive Environments Conference*, pages 203–209, 2021.

[7] Daniel R Berger, H Sebastian Seung, and Jeff W Lichtman. Vast (volume annotation and segmentation tool): efficient manual and semi-automatic labeling of large 3d image stacks. *Frontiers in neural circuits*, 12:88, 2018.

[8] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 2276–2279, New York, NY, USA, 2019. Association for Computing Machinery.

[9] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021.

[10] Kotaro Hara and Jon E. Froehlich. Characterizing and visualizing physical world accessibility at scale using crowdsourcing, computer vision, and machine learning. *SIGACCESS Access. Comput.*, (113):13–21, nov 2015.

[11] Lee Hinkle, Gentry Atkinson, and Vangelis Metsis. An end-to-end methodology for semi-supervised har data collection, labeling, and classification using a wristband. In *Workshops at 18th International Conference on Intelligent Environments (IE2022)*, 06 2022.

[12] Lee B. Hinkle, Gentry Atkinson, and Vangelis Metsis. Twristar - wristband activity recognition dataset. https://doi.org/10.5281/zenodo.5911808, January 2022.

[13] Seonghyeon Kim, Seokwoo Kang, Kwang Ryel Ryu, and Giltae Song. Real-time occupancy prediction in a large exhibition hall using deep learning approach. *Energy and Buildings*, 199:216–222, 2019.

[14] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, J. Macey, and M. Hebert. CMU Multimodal Activity Database (CMU-MMAC). http://kitchen.cs.cmu.edu/. Accessed on: May 4, 2024.

[15] Labelbox. The Labelbox Platform. https://labelbox.com/product/platform/, 2023. [Online; accessed 8-March-2023].

[16] Wagner Priscilla Madeo, Renata and Sarajane Peres. Gesture Phase Segmentation. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5Z32C.

[17] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[18] Project Jupyter. Jupyter Notebooks. https://jupyter.org/, 2023. [Online; accessed 8-March-2023].

[19] Dennis Stumpf, Stephan Krauß, Gerd Reis, Oliver Wasenmüller, and Didier Stricker. Salt: A semi-automatic labeling tool for rgb-d video sequences. *arXiv preprint arXiv:2102.10820*, 2021.

[20] Sameh Triki and Chihab Hanachi. A self-adaptive system for improving autonomy and public spaces accessibility for elderly. In *Agent and Multi-Agent Systems: Technology and Applications: 11th KES International Conference, KES-AMSTA 2017 Vilamoura, Algarve, Portugal, June 2017 Proceedings 11*, pages 53–66. Springer, 2017.

[21] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[22] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

[23] Zhaowei Zhu, Jialu Wang, and Yang Liu. Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. In *International Conference on Machine Learning*, pages 27633–27653. PMLR, 2022.

6 APPENDIX

ADDED/IMPROVED FEATURES

- **Refactoring and Modularization of Code**
  - Improved overall system performance by refactoring inefficient code and modularizing components based on design principles to enhance maintainability and scalability of the application.
- **Enhancements to Data Visualization Designs**
  - Implemented various user interface changes to facilitate easier and more intuitive data interpretation, reducing cognitive load and improving user experience.
- **Advanced Data Visualization Features**
  - Enabled click-drag functionality on graphs to dynamically fill user interface components with data values for manual labeling on the time-series plot.
  - Introduced synchronized visualization across all data representations, improving coherence between UMAP plots, time-series data plots, and video.
  - Enhanced label management by flagging uncertain labels for review based on confidence thresholds and allowing for subsequent unflagging after verification.
- **Expanded Dataset Testing**
  - Applied the tool to additional datasets, including the UCI Gesture Phase Segmentation dataset and the CMU Multimodal Activity (CMU-MMAC) Database, demonstrating the tool's adaptability to different types of activities and environments.
- **Local Deployment Enhancements**
  - Enabled local execution of the tool by allowing data and video to be loaded and processed locally, ensuring data privacy and reducing dependency on remote servers.
- **Centralized User Interface**
  - Developed a central interface that consolidates various functionalities including preprocessing, manual labeling, automatic labeling visualizations, and label correction into a single cohesive workflow.
  - This centralized approach provides a more streamlined and user-friendly experience, facilitating easier navigation and operation without the need for extensive technical knowledge.