Auto-curation of Large Evolving Image Datasets

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Sara Mousavicheshmehkaboodi

December 2021

Dedicated to my best friend and my love, Moa

Acknowledgments

First and foremost, I would like to thank my advisor, Dr. Audris Mockus. Throughout these years, Dr. Mockus has been much more than an advisor; he has been an incredibly supportive father figure to me. He has guided me, helped me see beyond what I could, and has championed me to pursue various directions in my research with guidance and careful nudges at every point in the way. I cannot thank him enough for his immense support throughout these years.

I would also like to thank the members of my committee, Dr. Hairong Qi, Dr. Amir Sadovnik, and Dr. Dawnie Steadman, for their invaluable feedback, insights, and ideas regarding my work.

Specifically, I would like to thank Dr. Steadman as well as my colleagues from the Anthropology department, Dr. Angela Dautartas, Megan Kleeschulte, Ileana Ilas, Tatianna Griffin, and Kelley Cross, for their wonderful collaboration. I have learned a great deal about human decomposition from them, and their help has been very influential in my PhD research.

A good support system is essential to having a smooth experience in graduate school in a foreign country. I like to thank my husband, parents, brothers, my sister Samira, and my in-laws, for their support in the past few years. I couldn't be here without their help. I am particularly grateful to my husband for his unconditional support and being the best mentor one could ask for. He taught me fishing instead of catching the fish for me and constantly challenged me to step out of my comfort zone, learn, and grow.

I would also like to thank my friends and colleagues, Eduardo Ponce, Mihaela Dimovska, Meghan McDonald, Dawn Sepehr, Sadika Amreen, Tapajit Dey, Yuxing Ma, Dylan Lee, and Zhenning Yang. They have been the nicest and most supportive friends and colleagues one could ask for. I would like to thank Eduardo and Mihaela for always encouraging me to work harder and appreciate the value of my work.

I will also be forever thankful to my Master's advisor, Dr. Chao Tian, who believed in me when I could barely speak English and gave me the opportunity to pursue graduate school and start this wonderful chapter of my life.

Abstract

Large image collections are becoming common in many fields and offer tantalizing opportunities to transform how research, work, and education are conducted if the information and associated insights could be extracted from them. However, major obstacles to this vision exist. First, image datasets with associated metadata contain errors and need to be cleaned and organized to be easily explored and utilized. Second, such collections typically lack the necessary context or may have missing attributes that need to be recovered. Third, such datasets are domain-specific and require human expert involvement to make the right interpretation of the image content. Fourth, the large size of these collections makes it time-consuming, costly, and in some cases, unfeasible to address the aforementioned problems. This dissertation aims to systematically address all four obstacles by curating (organizing, structuring, and enriching data in image collections). Specifically, we use a collection of 1M photos from forensic anthropology as well as other smaller image datasets to design and implement an auto-curation framework consisting of three overarching phases and associated unsupervised and semi-supervised techniques and tools to support each phase. As a result, we have developed data exploration techniques to support initial understanding of large image collections, an unsupervised clustering method for organizing such collections, a human-machine collaboration method to enable mass data labeling with relevant information, a semi-supervised method to reuse the existing expert-provided content for a small portion of a dataset and propagate it to the remaining uncurated data, and a system to preserve, publish and present the resulted curated data. Our evaluations of these techniques show that they outperform their corresponding state-of-the-art counterparts. The general auto-curation framework and tools presented in this work are applicable to any large image dataset, and the techniques are specifically designed for image datasets with evolving content. We employed the proposed tools and techniques for a large image collection of human decomposition in the forensic anthropology domain and, as a result, have enabled the use of digital resources for research where fieldwork is typically the norm. We hope that this work can help other disciplines to utilize the full potential of their data.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Operational data which are collected as part of an organization's day to day operations such as surveillance or customer support are increasingly used to gain more general insights beyond the objectives of the originating work process in numerous application domains such as banking, entertainment, health care, autonomous driving, remote sensing, cashier-less checkout systems, and many more [94]. Ready availability of such data has provided the opportunity for researchers and businesses to employ them for answering research questions, making high-stakes decisions and plans, and setting policies. However, the use of such operational data is fraught with risks as they may lack necessary context, have missing attributes, or be simply incorrect [94, 109]. Specifically, answering research and practical questions such as diagnosis from medical images or forming a research hypothesis requires a human expert typically trained in the domain to provide meaningful interpretations based on the data. Additionally, for researchers to interpret the data and be able to find desired content, the data needs to be organized, structured, enriched with proper attributes, and in one word curated for the tasks being solved [109].

However, such datasets often contain thousands or millions of items. The often large size of the unorganized data, combined with re-purposing it for new use cases makes the manual curation by domain experts and overall their involvement, costly and in cases infeasible, in turn hindering their efforts in interpreting and enriching the data. This introduces the need for auto-curation of the data in order to reduce the required human effort and cost using AI-driven methods and utilize the human effort to the maximum extent possible, resulting in an organized, contextualized, enriched, and overall curated dataset to support researchers in further interpretation and analysis in the domain. However to our knowledge, the literature lacks a well defined process for auto-curation of data that when employed for uncurated operational datasets can lead to analytical data suitable for answering various research and business questions.

In this work, we introduce an auto-curation framework that recognizes three distinct phases: 1) initial data acquisition, exploration, cleaning, and organization (intake), 2) contextualization (enrichment), and 3) providing support for preserving, publishing and presenting the curated data (dissemination).

Throughout this work, we focus on large image datasets, which are often created through operational processes, manual collection, or web scraping and are rarely organized [19, 143, 118], free of error, or contain context or content attributes needed to conduct research, and in short, are seldom curated. As with any other operational data, the mismatch between what data has and research needs may be explained by the following reasons: a) the final usage of such datasets often not being fully defined at the time of data collection, b) data having been collected for one goal and later repurposed for another, and c) the data being a result of integrating multiple data sources initially collected for different reasons and through different means [118, 94].

Although the presented auto-curation framework's components are general and applicable to other image datasets, the techniques used to automate the curation tasks and their implementations are data-dependent. There are various types of image datasets used for many applications, such as facial recognition, action recognition, object detection and recognition, handwriting and character recognition, and scene understanding. Images representing evolving phenomena, such as images of human decomposition, aging faces, growing plants, or decaying produce, are one specific type of image dataset. In such datasets, in addition to having various views of a given subject, which is often the case in image datasets, there also exists a time component. In such datasets, what is captured in the images evolves through time, and even though a set of images represent the same object, their appearance may be drastically different. An illustration of the structure of such image data is shown in Figure 1.1. Presently, there is a lack of understanding of how latent relationships

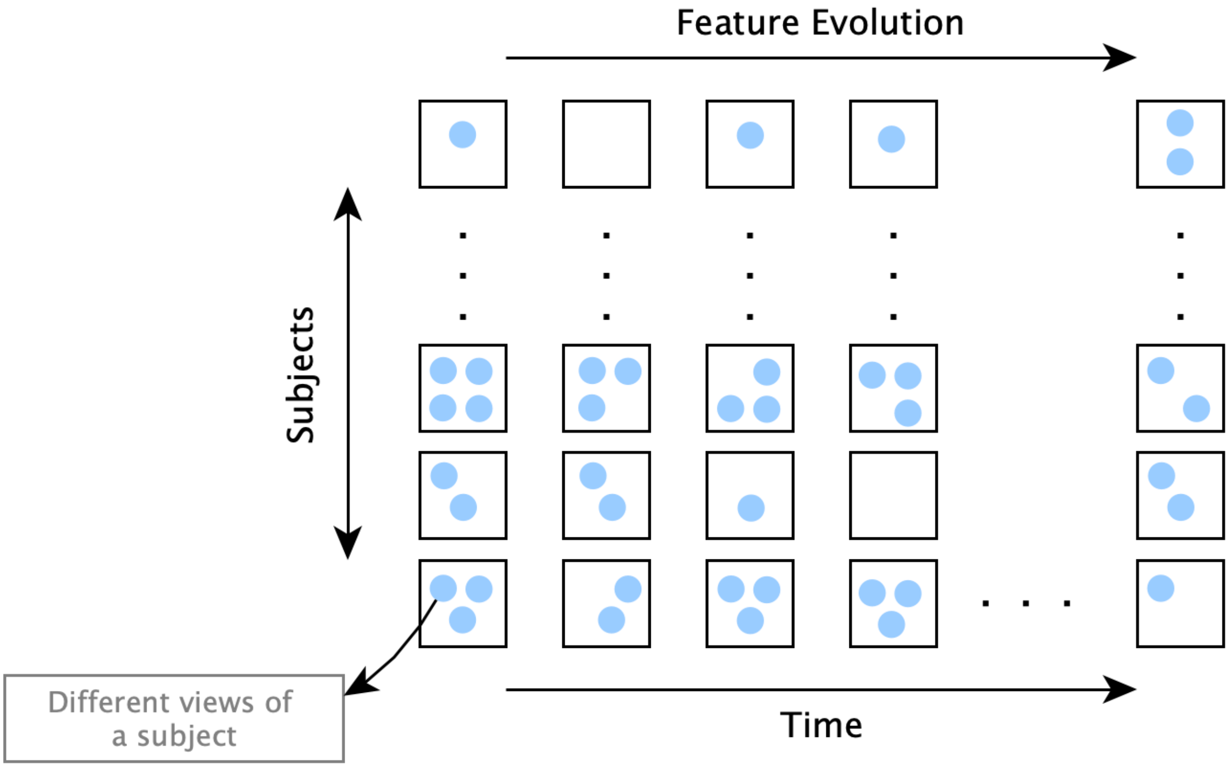


Figure 1.1: An illustration of the structure of image datasets with evolving content

in such collections may present challenges or opportunities for the auto-curation process of a dataset. This dissertation specifically focuses on such image data.

The key contribution defended in this dissertation is proposing an auto-curation framework for large image datasets representing evolving content and the implementation of it, which includes:

1. Defining curation tasks for large image datasets and implementing these tasks in the context of image datasets representing evolving content

2. Utilizing the characteristics of evolving datasets to design and develop unsupervised techniques for performing the organization task of the curation without human involvement

3. Developing a human-machine collaborative technique to minimize the curator's efforts in the curation process and simplify mass labeling

4. Designing and developing an AI-based technique to reuse and propagate the limited number of expert-provided labels to a large portion of the dataset by utilizing the characteristics of evolving image data

5. Enabling future usability and further enrichment of the curated content by providing a platform to support collaborative and crowdsourced efforts to iteratively refine the curated content as well as preserve, present, and publish the curated data

In the following, we first provide our definition for data curation, and then details about curation tasks and our developed techniques to support them.

1.1 Data Curation

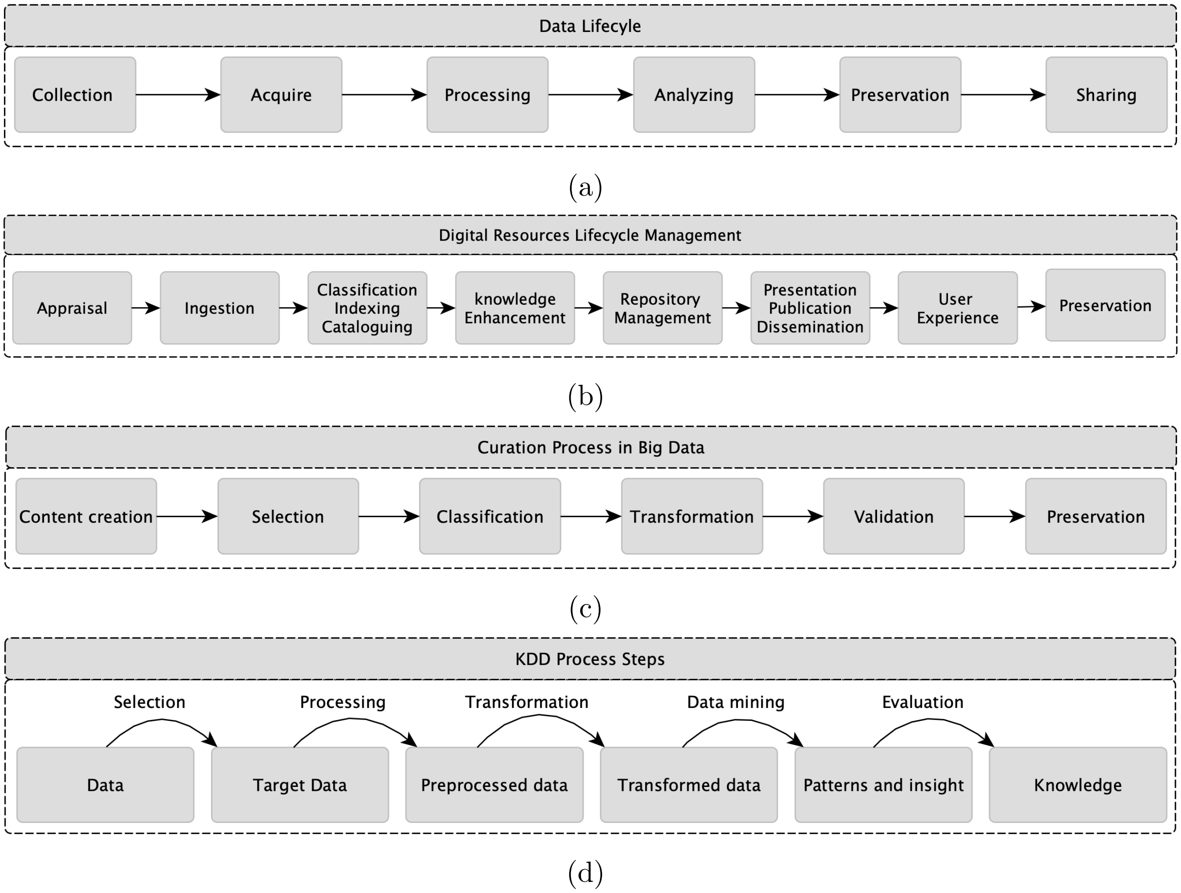
In the field of information science, the term *Data Curation* is defined as an active and ongoing management of data through its lifecycle of interest and usefulness [31]. A data lifecycle model is presented by Cragin et al. which includes collection, acquiring, processing, analysis, preservation and sharing of the data as shown in Figure 1.2a. Similarly, Constantopoulos and

Dallas have presented the components for the lifecycle management of digital resources as a whole [25]. These components are appraisal, ingestion, classification, indexing, cataloguing, knowledge enhancement, user experience, repository management, presentation, publication, dissemination, and preservation as shown in Figure 1.2b. These components are geared to achieve a) trustworthiness of digital resources, b) organization, archiving and long-term preservation, and c) adding value to the resources [25], similar in spirit to the data lifecycle model.

Within the area of big data, Freitas and Curry mention data curation as an important component in the big data value chain and state that data curation processes can be categorized into different activities such as content creation, selection, classification, transformation, validation, and preservation as shown in Figure 1.2c [49].

Furthermore, curation is also attended to in the area of Knowledge Discovery in Databases (KDD). The KDD "process" is the operation of using a database along with any required selection, preprocessing, sub-sampling, and transformations which effectively make up the curation step. In the KDD process, the transformed data is then fed into data mining methods to extract patterns. The products of data mining are then evaluated to identify the subset of the extracted patterns deemed "knowledge" [47]. The steps of this process are shown in Figure 1.2d.

However, in all areas mentioned above, there is a theme for what data undergoes as categorized by colors in Figure 1.3. First, all areas require finding and obtaining the desired data, assessing what part of it is to be used, ingesting, cleaning, and storing it for further usage, modifications, and analysis in the future. Second, the data often needs to be converted from its raw format and combined with other sources of information to be contextualized, enriched and consequently made ready for statistical analysis and developing predictive models. Therefore, a processing step is often required. Third, the value of the curated data is only realized if shared and used by researchers and end-users. Hence, the data needs to be preserved and disseminated. In this work, inspired by the past definitions and usages of data curation process, we define data curation for images as the process of turning unstructured or semi-structured operational image data collections into unified, cleaned, organized, contextualized, and enriched datasets ready to be utilized by end-users



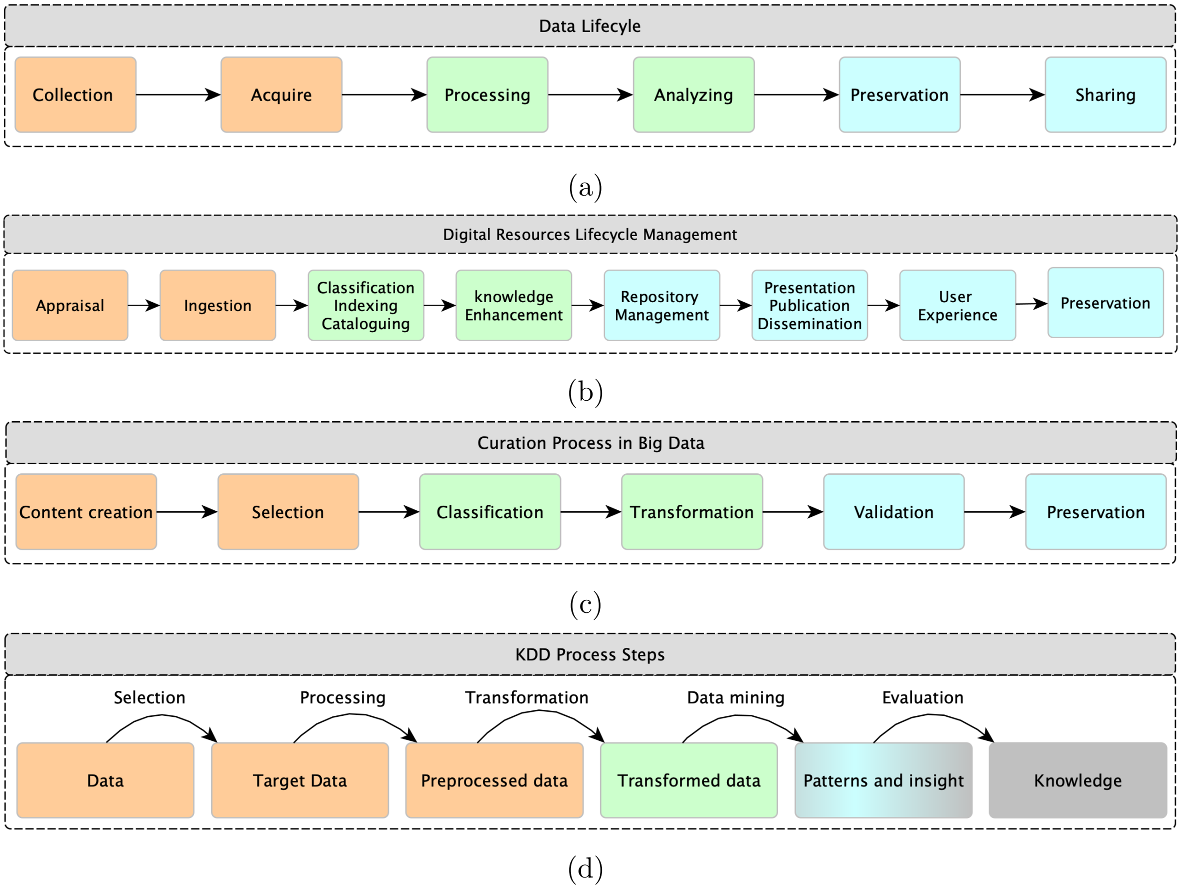
(a)

(b)

(c)

(d)

Figure 1.2: The lifecycle of research data, components of digital resources lifecycle management, curation processes in big data, and steps comprising the KDD process are shown in (a), (b), (c), and (d) respectively.



(a)

(b)

(c)

(d)

Figure 1.3: The lifecycle of research data, components of digital resources lifecycle management, curation processes in big data, and steps comprising the KDD process are shown in (a), (b), (c), and (d) respectively. The colors are used to show high level steps of what data undergoes in the curation process per this work's categorization. In the KDD process, "knowledge" is shown with a different color due to the fact that it is not a curation step and is rather enabled by curation and obtained through mining and analyzing curated data. "pattern and insight" is also shown in a different color due to the fact the the curation itself does reveal some patterns in the data but further analysis enabled by curation can also be done after the curation phases to extract further insight about the data.

and researchers for various applications. We define three main overarching phases for data curation in the context of evolving image collections that can provide support for the data throughout its lifecycle and present AI-assisted techniques and tools to support each phase. The three phases are 1) intake, 2) enrichment, and 3) dissemination and are described in more detail in the following sections. An overview of the presented auto-curation framework is shown in Figure 1.4.

1.2 Curation Phases

To achieve the goal of producing a curated dataset, we define three overarching steps of data intake, enrichment, and dissemination to increase the quality of the data in terms of structure and format, the inclusion of relevant domain information, and allowing re-usability and future improvements on the data.

1.2.1 Intake

What a curated dataset is expected to be, by which the detail of the curation phases is determined, heavily depends on its use-cases. What is of interest in an image dataset varies depending on the ultimate applications of it. For instance, if the task at hand is to classify images based on their content, then image-level labels are required. However, if one is interested in not only the type of objects depicted in each image, but also the exact location of them in the image, then pixel-level labels are required.

Generally, any collected data undergoes several ad-hoc iterations of enhancement through using that data for different purposes. Such iterations result in producing organized, structured and enriched data that is ready and sufficient for answering various research questions. One example of such data is the PASCAL VOC dataset [44] that can be used for various computer vision tasks such as classification, classification with localization, and segmentation. To transform a raw image dataset to a format suitable for various tasks without performing multiple iterations of data organization and enrichment, it is crucial to first identify the main potential domain use-cases for the data and define a valid nomenclature

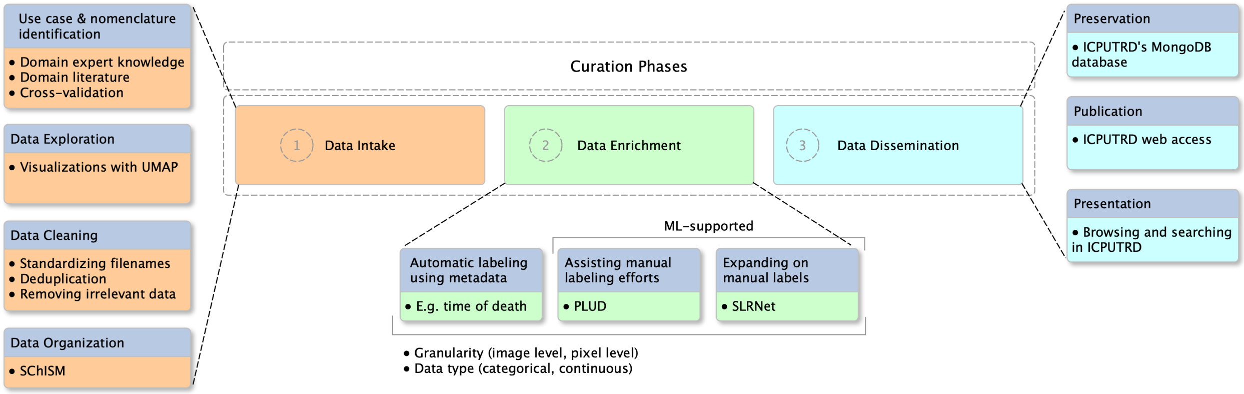

Use case & nomenclature identification
• Domain expert knowledge
• Domain literature
• Cross–validation
Data Exploration
• Visualizations with UMAP
Data Cleaning
• Standardizing filenames
• Deduplication
• Removing irrelevant data
Data Organization
• SChISM
Curation Phases
1 Data Intake
2 Data Enrichment
3 Data Dissemination
ML–supported
Automatic labeling using metadata
• E.g. time of death
Assisting manual labeling efforts
• PLUD
Expanding on manual labels
• SLRNet
• Granularity (image level, pixel level)
• Data type (categorical, continuous)
Preservation
• ICPUTRD's MongoDB database
Publication
• ICPUTRD web access
Presentation
• Browsing and searching in ICPUTRD

Figure 1.4: Different components in the presented auto-curation framework for evolving image datasets.

accordingly. This step helps with detailing the next set of curation tasks according to which the desired content to contextualize and enrich the data with, is identified.

Data intake includes the process of finding the desired data, assessing the usefulness of various subsets of it with respect to the identified use-cases and nomenclature for the domain, and moving it from one or more sources to a destination where it can be stored and cleaned. In this phase, the data undergoes a general exploration to identify what it holds, what is of interest and what is irrelevant, as well as general cleaning, and organizing.

1.2.2 Enrichment

An essential curation phase is data enrichment, by which the dataset is supplemented with additional relevant information representing the content of each item in the dataset. This consequently enables future searching, querying, and analyzing any desired set of images with specific characteristics. Such meaningful information could be obtained from existing metadata, external datasets, manual labeling by human annotators or domain experts, or expanded expert-provided input using machine learning algorithms.

However, in cases that human intervention is required to obtain labels, due to the often large size of image datasets and the cost of manual labeling, fully manually obtaining these labels is not feasible. Instead, machine learning and computer vision algorithms such as image classification and segmentation methods can be used to label the data with different levels of supervision in the process, assist users in the labeling process, and expand a small set of image-level or pixel-level labels provided by human annotators to the entire dataset in an automated manner. However, the off-the-shelf state-of-the-art computer vision algorithms that could potentially be used for this purpose often do not perform well on image datasets with evolving content, or their performance could be improved by taking advantage of the unique properties of evolving image datasets.

The specific characteristic of evolving image datasets, gradual changes in the appearance of the images depicting the same subjects through time due to evolution, results in both challenges and opportunities for computer vision tasks. The challenging scenario is caused by drastic differences in appearance for the images with the same semantic at distant timesteps in the evolution timeline, posing challenges for state-of-the-art similarity-based methods such as clustering and classification and making them perform poorly. On the other hand, the gradual changes in such datasets result in a local similarity between images of the same subjects in neighboring timesteps in the evolution timeline, creating opportunities for linking images of the same content together as well as reusing human-provided labels for a subset of images and propagating them to the unlabeled images which results in providing more training data that can lead to higher performance for computer vision techniques.

1.2.3 Dissemination

An organized, contextualized, and enriched dataset can only be utilized if it is preserved and shared with potential end-users to be used and even further enriched if desired. Therefore, the last phase of our presented auto-curation framework is data dissemination, in which the preservation, publication, and presentation of the curated data are handled. In other words, the dissemination phase is concerned with the preservation and management of the curated data, allowing users to access and utilize the curated content, and interact with the data through searching, browsing, and producing additional new information if desired.

To enable data dissemination, a platform is needed to preserve the data and facilitate access to, the use of and interaction with the data through searching for a desired set of images, browsing and exploring images, and adding new labels to the data by various end-users.

1.3 Datasets

In this dissertation, we apply our proposed auto-curation framework on image datasets with evolving content. In such datasets, the objects of interest are photographed with varying time intervals to capture and study their changes through time. These datasets might also be collected for archival purposes and later repurposed for other applications. In the following, we describe a few evolving image datasets used throughout this dissertation to evaluate our proposed techniques for the purpose of auto-curation.

1.3.1 Human Decomposition

Images tracking human decomposition are of great value for research in the area of forensic anthropology due to a few main reasons. First, when using images instead of actual decaying bodies, studying the decomposition process is no longer limited to only the period of decomposition itself. Second, it allows researchers to focus on particular factors and correlate them to environmental and individual characteristics (e.g., age or weight). Finally, it enables testing hypotheses and studying the decomposition process using a large sample size of subjects.

The type and level of decomposition are not the same for various subjects or even various parts of a given subject. Studying human decomposition helps define which environmental and individual factors affect the rate of decay, which can help develop new methods to estimate postmortem interval (time since death) and help investigators evaluate death scenes.

Overall, there are five stages of decomposition that human and animal remains go through; "autolysis", "bloat", "active decay", "advanced decay", and "skeletonization". The initial breakdown starts as soon as death occurs due to lack of blood circulation and oxygen flow, resulting in muscular tissues becoming rigid and the blood pooling into the lower areas, and the bacteria in the intestines starting their activities. This stage happens inside the body and is not visible from the outside. At the end of this stage, blowflies and flesh flies arrive to lay eggs. In the next step, the body is bloated due to the activities of the bacteria, which produce gasses such as methane, carbon dioxide, nitrogen, and hydrogen sulfide. These gasses cause pressure and result in pushing fluids out through natural openings and other openings that may have been caused by maggots feeding on the body tissues. The next step is the "active decay" step, where a lot of fluid is released from the body, resulting in a large body mass loss and attracting more insects. In the "advanced decay" step, most soft tissues have already decomposed, and only bones, hair, cartilage, and ligaments are left, which attract beetles and certain types of flies. They feed on the remaining tougher material. Finally, in the last stage, mites and moth larvae digest any remaining hair, and only the skeleton will be left.

The human decomposition image dataset used in this work is collected by taking photos of decomposing humans donated to the Forensic Anthropology Center at the University of Tennessee. Subjects are placed into the Anthropology Research Facility known as the "Body Farm" where the different stages of decomposition are studied. The photographers who take these pictures have a protocol to follow to capture all portions of the body. However, due to different body placement positions and the changing of photographers over the years, the content of the photos is always changing and difficult to predict. The photos are taken periodically from various angles to show different stages of body decomposition. The collection spans from 2011 to 2017 and has over one million images. The image resolutions vary from 2400 × 1600 up to 4900 × 3200. The photos are stored based on an ID associated with the subject and the date of the photograph.

In this dataset, throughout the decay process, bodies undergo many changes and evolve through time. Figure 1.5 shows an example of tracing a body part throughout its decay process.

1.3.2 MORPH

The MORPH Database [111] contains mugshots collected over a span of 5 years with images of the same subject. The images are taken in real world conditions and not in controlled environments. The dataset also contains metadata in the form of age, gender, and race. The MORPH dataset includes $55,134$ images of $13,618$ subjects ranging from 16 to 77 years old. Figure 1.6 shows various mugshots from a subject at different ages from 38 to 48.

1.3.3 Aberystwyth Leaf Evaluation

Aberystwyth Leaf Evaluation dataset is released by Aberystwyth University. The dataset has been collected to support their work titled "Dynamic Modelling of Plant Growth with Computer Vision". This dataset has been released to be used by researchers to further advance state-of-the-art methods used in image analysis for plant sciences. The images are collected by periodically taking pictures from the Arabidopsis plant with 15-minute intervals using a robotic greenhouse system. The Aberystwyth dataset includes manual annotations for a subset of the images [9].

In this dataset, four sets of 20 Arabidopsis Thaliana plants have been grown in trays. The total number of images is 6702 (134040 individual pots), in which there are 56 annotated ground truth images containing 916 individual Arabidopsis plants. Figure 1.7 shows an example of growing plants. This example illustrates a gradual change in the appearance of the plants due to evolution and growth.

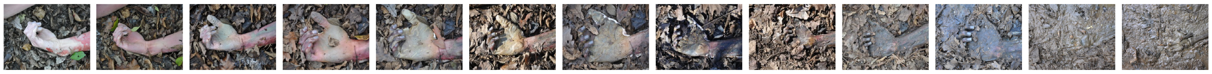


Figure 1.5: An example of evolution in the human decomposition dataset.

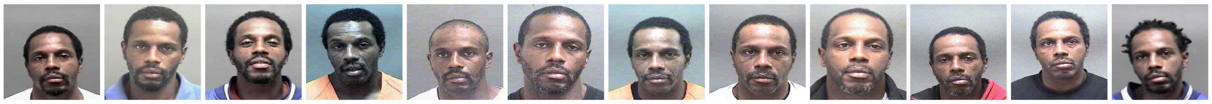


Figure 1.6: An example of evolution in age progression in the MORPH dataset. Photos show a subject from age 38 to 48. Multiple images may exist for a given age.

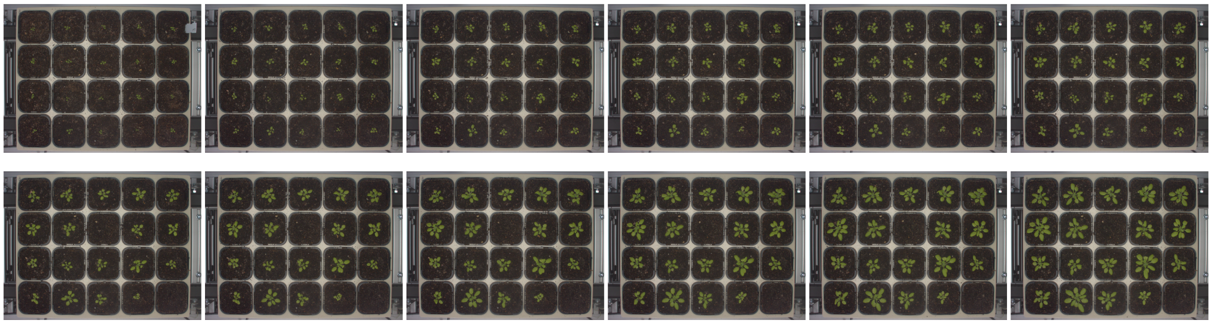


Figure 1.7: An example of evolution in growing plants in the Aberystwyth dataset.

Chapter 2

Review of Literature

In this section, the related works to the overall topic of this dissertation, auto-curation, and the techniques developed to support the presented auto-curation framework, namely image clustering, classification, segmentation, and tools, are provided.

2.1 Auto-curation

The concept of curation is defined in many domains. For example, in the field of Information Science, it is defined as active and ongoing management of data through its lifecycle of interest and usefulness, and is a part of the solution for data preservation and sharing to promote cross-disciplinary reuse and discovery [31, 137, 30, 64, 26, 29]. Nowadays, with the availability of a large amount of data, curation has become an essential component in the big data chain [84, 126, 49] and is comprised of content creation, selection, classification, transformation, validation, and preservation. In these areas, curation is mainly used when combining and integrating data from multiple resources and as the process of turning independently created data sources into unified datasets ready for analytics through a process guided by domain experts to present the best content available.

With the advent of large image datasets, collected from web scraping or other sources such as the Landsat program [118, 143, 19, 139], and their various use cases in businesses and research domains, the need for high quality and research- and analysis-ready format data, in a nutshell, curated data, is felt more than ever before. A few works have been presented to curate such datasets by proposing cleaning methods that can result in organized data with better labels [42, 149, 83].

However, in all aforementioned works and areas, the focus has been mainly on the multi-source integration and cleaning aspect of curation when working with multiple data sources or cleaning and improving existing labels when working with a single image dataset. To the best of our knowledge, there is no definition or framework that lays down a set of curation tasks for a raw image dataset or, for that matter, for a raw image dataset with evolving content that can cover all stages from ingesting the data, processing and transforming it into a format suitable for research and analysis, to using the data and sharing it with others while allowing for further improvements on the quality of the curated data.

In this work, we present an auto-curation framework for large image datasets with evolving content. This framework includes three phases for curation (ingestion, enrichment, and dissemination) that, when employed on an uncurated image dataset, lead to an organized, contextualized format suitable for research and analysis. The main machine learning concepts employed and the specific methods developed and implemented to perform the curation tasks for large image datasets in an automatic or semi-automatic manner are image clustering, classification, and segmentation reviewed in the following.

2.2 Clustering of Evolving Images

Organizing and categorizing data is an essential part of auto-curation that can help produce searchable and browsable data while facilitating data labeling. Image clustering can be used to organize a dataset in an automated manner without human involvement. Unsupervised image clustering is the technique of grouping images with similar characteristics without any supervision or prior knowledge of their actual labels. Using unsupervised methods, one can cluster and categorize images using their numerical embeddings in groups that share similar characteristics [56, 145, 99, 134].

An evolving image dataset can be clustered based on various aspects such as stages of evolution or the conceptual object/objects depicted in the images. Multiple Clustering methods have emerged as a result of seeking alternative clusterings that group a given dataset into clusters that exhibit different aspects of similarity. The works in [7, 103, 146] build clusters based on dissimilarity and the quality of the clusters, by forcing new clusters to be different than existing ones. In the meta-clustering method presented in [21], several alternative clusterings are found so that users can decide what set of clusters fit their need best. Similarly in [63], authors find multiple clusterings by minimizing the correlation between them through an objective function. In [33], alternative clusterings are obtained by maximizing the likelihood of each of the alternative clusterings over the data, while minimizing the similarity between them. In all of these methods, each set of clusters is based on a single criterion. However, in the case of an evolving image dataset, we aim to jointly cluster the data based on two criteria: the concept of objects depicted in the images and the concept of the evolution stages.

Related to the aforementioned need are multi-view clustering methods. Multi-view clustering emerged from attempts to cluster objects based on their semantic and conceptual similarities even though they might have different appearances [144, 59]. Although it might seem that we can map the problem of clustering evolving image datasets to multi-view clustering, there is a fundamental difference that make multi-view methods less suitable for such cases. In the case of evolving image data, ideal clusters that include images of the same object with multiple views also evolve over time due to evolution. Thus, the same view of the same object appears differently depending on the stage of evolution.

Recent work has also explored the combination of image clustering and deep representation learning [99, 131, 56]. Guérin et al. [56] studied the effect of using feature representations obtained from pre-trained convolutional neural networks (CNNs) on image clustering and showed that using feature representations obtained from such networks results in better quality clusters.

Other works [145, 135, 20, 20] present end-to-end methods for unsupervised feature representation learning of images. Yang et al. proposed a recurrent framework for joint unsupervised learning of deep representations and image clusters by leveraging the fact that good representations are beneficial to image clustering which can be used to supervise the representation learning process [145]. In another work [135], authors trained a task-specific deep architecture for clustering. In DeepCluster, Caron et al. [20] present an end-to-end method that consists of a collaboration between clustering and classification for feature representation learning in large scale datasets in an unsupervised manner.

In our presented work for clustering evolving image datasets [98], we utilize deep learning representations as in the above. However, these methods cannot address the implicit constraints imposed by the temporally evolving objects alone.

Our method is partially inspired by techniques in the area of data stream clustering, which is used to monitor, for example, energy consumption, financial transactions, industrial sensor data, urban traffic and live update of stock trading. Stream clustering deals with large amounts of data that cannot be stored in memory and thus random access is not possible. Algorithms used for this purpose handle the evolution and changes in the number of clusters as new batches of data come in. One common approach in stream clustering is to use a sliding window, introduced by Aggarwal et. al [5], to keep track of how cluster centers change as new data points are streamed into the algorithm. Aggarwal et. al [5] used a sliding window instead of one-pass clustering to provide a better understanding of evolving behavior of the clusters. Several other methods [61, 150] have been built on this idea to improve the efficiency and accuracy of stream clustering.

However, the problem of clustering images with evolving content cannot be directly mapped to stream clustering. In the case of evolving objects, not only does the number of clusters vary from one observation to another, but also the overall object representations change dramatically through time. Inspired by past works, we use a sliding window along with a dynamic inclusion criteria to build sequences of evolving images belonging to the same class to form clusters (further detailed in Section 3.4).

2.3 Classification of Evolving Images

Unsupervised methods such as image clustering can organize the data and provide implicit labels for an image dataset by grouping similar images together. However, to search for images or perform analysis based on their content, they need to be labeled with proper keywords representing what they depict. Therefore, labeling image datasets is essential and enables the useability of such datasets in various applications such as studying environmental changes, land use and land planning, urban planning, surveillance, geographic mapping, disaster control, and object detection. Additionally, labeling images with relevant information is a crucial step in auto-curation as it extends the dataset with searchable labels that can also be used for data analysis.

LabelMe [115] and similar interfaces [28, 97, 140, 151] have been used to label image collections. The human labeler needs to provide labels for the images manually, using these methods. Given the size of image collections and the amount of labeling time required, manually labeling a large image dataset is costly, time-consuming, and unfeasible.

Fluid Annotation [6] assists the labeling effort by providing the initial labels that can be edited as needed. Fluid Annotation uses Mask-RCNN [57] as the primary deep learning model. For Mask-RCNN and other deep-learning-based techniques such as Deeplabv3+ and YOLO [23, 110] to work, large, complete, and clean training datasets such as Open Images, ImageNet and COCO [71, 37, 17] are required. Such approaches without additional training do not work for a dataset with a completely different set of object classes.

Well trained classification models can be used to generate labels for unlabeled images. The common approach for building a classification model is through a supervised approach, where a set of labeled training data is available and used as supervision in the model training process. However, good, representative training data is not readily available, and there are many large unlabeled datasets in various domains where manual labeling is time-consuming and expensive. This problem is exacerbated when labeling such datasets require domain expert knowledge and, due to the scarcity and public unavailability of such domain-specific data, there is no similar labeled data available that can be utilized for transfer learning purposes. Therefore, the scarcity of clean labeled data hinders the performance of supervised models.

On the other hand, active learning-based models [121, 117, 153, 46, 27] start with a small set of labeled data and gradually improve their performance. They are used to iteratively provide more training data from the unlabeled pool of images and improve the classifier's performance. The classifier is first trained on the seed (the initial small batch of labeled data). Then in each iteration, new samples of the unlabeled pool that have a better potential to help with the learner's learning process are selected and labeled by an "oracle". Fisher et al. used a similar approach and presented an iterative work with a human in the loop for labeling a large dataset [148]. Their work includes iterations between a classifier and manual human annotations of sampled data to generate training data for the next round of training.

In this work, we present a technique similar to the active learning models in spirit but leveraging semi-supervised assumptions to iteratively increase the size of the training data for the classifier and improve its performance while reducing the human labeling effort. Semi-supervised methods are based on three main assumptions: 1) smoothness: if two images are similar, their labels should be the same, 2) low density: class boundaries should not pass through high-density regions, and 3) manifold: data points on the same low dimensional manifold should have the same label. These assumptions are realized in our method in the form of a semantic clustering method designed for evolving data where similar data points that could fall in the same cluster tend to have similar labels. In iterative methods with a human in the loop, this concept can be used to reduce the effort of the human annotator in the labeling process. Inspired by this observation and active learning models, we introduce and develop a platform for labeling unlabeled datasets called PLUD that enables mass data labeling in an iterative, semi-supervised manner by building a collaborative and iterative system between clustering, human annotator, and classification.

2.4 Segmentation of Evolving Images

Image segmentation entails segmenting an image into different classes and essentially labeling each pixel with the class it represents. It has a wide range of applications such as image segmentation for road-driving images [132, 34] and medical images [120, 107, 85]. It is an important technique for analyzing what's inside an image and can help study the co-occurrence of events in images. We use semantic segmentation as part of the enrichment phase in our auto-curation framework to enrich image datasets with more detailed labels than image-level labels.

There are various types of image segmentation namely, object localization [54], semantic segmentation [10], instance segmentation [68, 88], and panoptic segmentation [69, 77, 142, 78]. Object localization methods locate objects in images with a bounding box around each object. Semantic segmentation is referred to as the task of identifying different classes of objects in an image. It broadly classifies objects into semantic categories such as 'person', 'book', 'flower', 'car', etc. Instance segmentation segments different instances of each semantic category and can be viewed as an extension of semantic segmentation. For instance, if a semantic segmentation method identifies pedestrians crossing a street in an image, then instance segmentation identifies individual (instances of) people in the image. Panoptic segmentation is the combination of semantic and instance segmentation techniques. It semantically distinguishes different objects and identifies separate instances of each kind of object in the input image. The name 'panoptic' is used because it enables having a global view of image segmentation (category-wise and instance-wise).

Images with evolving content usually aim at capturing and tracking the evolution of individual subjects. As a result, there are not many instances of the same class in each image. Therefore semantic segmentation seems to be sufficient to understand what classes exist in an image and where in the image they are located.

Similar to other supervised computer vision methods, supervised semantic segmentation requires many training labeled images to perform well. The limitations of supervised learning are most pronounced in the task of image segmentation [48] because it requires pixel-level labels. Labeling and annotating images is a time-consuming and challenging task in general, with a single image taking from 19 minutes [17] to 1.5 hours [28] on average. Human annotation of static images for segmentation is particularly expensive, requiring, for instance, 90 minutes per image [28] or 22 worker hours per 1,000 pixel-level labels [81].

To diminish the cost of image labeling, semi-supervised and weakly supervised (e.g. data with labels at a coarser granularity than those needed) methods have been developed. Semi-supervised and weakly supervised learning [40, 90, 113, 147, 75, 55, 136, 72, 52, 80, 65] target speeding up the data labeling process and other computer vision tasks by leveraging a large amount of available unlabeled data and using weakly labeled data. These approaches start with a small set of labeled data and a large set of unlabeled data. The labeled data then guides the learning process to make the models more generalizable to the remaining unlabeled data.

Various methods [125, 60, 104, 154, 152] have been developed that achieve state-of-the-art on benchmark datasets such as VOC12 [44] and COCO [81] as well as domain specific data such as medical images [36, 108].

Generative Adversarial Networks (GANs) have also been used in a semi-supervised learning setting [125, 60] by generating adversarial examples using GANs for semantic segmentation and extending the generic GAN framework to pixel-level predictions. However, samples generated by adversarial-based methods may not be sufficiently close to real images or labels to help the segmentation network; and other non-GAN-based methods have shown better performance [104, 154].

Consistency-based training has also been used for semantic segmentation with promising results. For example, CCT [104] is based on cross consistency learning for semantic segmentation and uses a segmentation network with an encoder and a decoder for annotated images. Additionally, CCT adds several auxiliary decoders that use perturbed versions of the encoder's output for the unlabeled data. It then enforces consistency over the outputs of the auxiliary decoders and that of the main decoder. CCT has achieved state-of-the-art results on the VOC12 [44] dataset, outperforming the aforementioned adversarial based methods [60, 125].

Other methods, such as PseudoSeg, combine ideas of pseudo-labeling and consistency learning [154]. The authors use a similar idea to consistency-based methods to generate pseudo-labels. They fuse a self attention-based GradCAM of an unlabeled input image to their network's prediction for a weakly augmented version of that same input and use the result as a pseudo-label for the input. Then, they enforce consistency between the predictions of the network for a strongly augmented version of that input and the pseudo-label.

In this dissertation, we present a simple method that does not rely on external augmentation and perturbation and is conceptually simpler. The key idea of our method is to find a way to effectively reuse the existing differences and similarities in the dataset itself.

2.5 Data Interaction and Labeling Tools

Once a dataset is prepared with the right format and is enriched with the right content, in other words, it is curated, it is only valuable if end-users and researchers use it. Therefore, it is important to have a system or a platform that enables preserving, publishing (allowing multiple users to access), and presenting (allowing users to use the data by searching for their desired content, querying, browsing, and labeling) the curated data. Many labeling tools have been developed to assist the manual labeling process of large image datasets [115, 28, 97, 140, 151, 6]. However, these systems only provide support for labeling and do not allow the use and preservation of and interaction with the curated data. In this dissertation, we present a web-based platform called ICPUTRD built using the MEAN stack to enable the preservation, publication, and presentation of a large, curated image dataset.

Chapter 3

Data Intake

Figure 3.1 shows the components of the data intake phase. Data intake includes the process of finding the desired data, assessing the usefulness of its various subsets with respect to its use-cases, and moving it from one or more sources to a destination where it can be stored, cleaned, and prepared for future use. In this phase, first, the use-cases and a proper domain-relevant nomenclature are identified and decided. Next, data is stored and undergoes a general exploration to identify what it holds, what is of interest, and what is inconsistent, corrupted, or irrelevant. Then the data is cleaned according to the information gained from the exploration phase. Finally, the data is organized.

In the following, we provide details on use-case and nomenclature identification, data exploration, and data cleaning. Since other image datasets introduced in Section 1.3 are already curated, we use the human decomposition dataset to illustrate the tasks done in these steps.

3.1 Use-case and Nomenclature Identification

To transform a raw image dataset to a format suitable for various tasks without performing multiple iterations of data organization and enrichment, it is crucial to first identify the main potential domain use-cases for the data and define a valid nomenclature accordingly. This step helps with detailing the next set of curation tasks according to which the desired

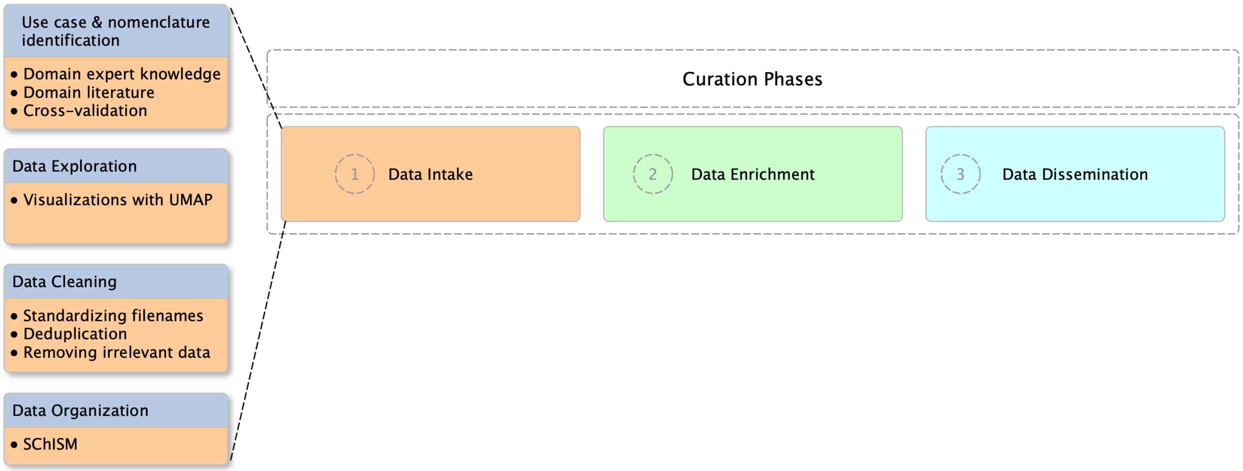


Figure 3.1: Data intake components.

content to contextualize and enrich the data with, is identified. In addition, using a standard nomenclature enables providing meaningful benchmarks and comparisons.

Identifying the potential use-cases of a domain-specific dataset and developing a proper nomenclature to be used in its curation process can be done through a few main procedures:

1. Meetings with domain experts and the beneficiaries for that dataset
2. Reviewing domain literature
3. Cross validating labels provided by multiple domain practitioners

3.1.1 Domain Expert Knowledge

In the case of the human decomposition images used in this work, after a few iterations of discussing the potential use cases of the dataset with experts in the Forensic Anthropology domain through regular meetings, it was confirmed that the end users of this dataset may be forensic researchers, law enforcement officers, students, or teachers.

3.1.2 Domain Literature

In order to enrich the data with information suitable for all the potential users, we developed a standard nomenclature. The nomenclature is based on the amalgamation of widely acceptable terminology, common between law enforcement and academic researchers, from the forensic literature on human decomposition. To maximize usability, a set of self-explanatory terms (e.g. maggots, mummification) are used.

3.1.3 Cross Validation

The finalization of the nomenclature was done by asking expert users to label a subset of images using keywords from the compiled nomenclature that most accurately described the content and the decomposition stage depicted by them. The keywords resulting from this step were then cross validated. Keywords that occurred most frequently were prioritized in terms of developing a consistently applied terminology. After an initial round of labeling, the labels were reviewed to ensure that the terms were being applied consistently. If multiple terms were being used to describe the same feature, one term was chosen for simplification (e.g. “mummification” was chosen among itself and other descriptions such as “mummified”, and “mummified skin”). A list if the developed nomenclature is shown in Table 3.1.

3.2 Data Exploration

Data exploration is an initial step in making sense of the data. In addition, it is a crucial step for a proper, effective, and efficient data cleaning which in turn directly affects the quality of the data. Data exploration of a large image dataset is done toward answering questions such as the following:

- How many images are in the dataset?
- How many categories are in the dataset?
- How many images exist for each category?
- What is the aspect ratio of the image sizes?
- What are the outliers?
- What is the irrelevant and undesired information?

Answering these questions is essential for cleaning the data as well as selecting proper techniques as well as balanced and suitable samples for further data analysis. However, manually browsing through a large image collection for answering such questions can be very time-consuming. Therefore, other visualization tools and scripts are needed to facilitate this process.

In this work, we used *Python* and *Bash* scripts as well as machine-learning-based visualization tools to visualize the relationship between the images in their embedding space. Examples of using these tool for exploring the human decomposition dataset is provided in the following.

In the human decomposition dataset, we explored the data to answer the following questions based on discussions with domain experts:

Table 3.1: The nomenclature developed for labeling images of human decomposition

List of keywords				
adipocere	brown discoloration	glossy skin	mesh bag	saturated soil
advanced decomposition	bruising	gray discoloration	moist decomposition	scale
amputation	bulla	green gut	moist skin	scar
ant	bullae	groin	mold	scatter
ants	cage	hair	mottling	scavenging
arm	clothing	hand	mouse	shoe cover
autopsy bag	darkened skin	head	mouth	skeletonization
autopsy cut	dentures	hernia	mummification	skin slippage
axilla	dermis	hips	mummified	skull
back	desiccated skin	insect	neck	snail
background	desiccation	larvae	nose	soft tissue
backside	discoloration	larval mass	open sore	soil stain
bandage	dry skin	leaf	organ bag	stake
bee	duct tape	left foot	personal effects	stink bug
beetle	duplicate	left hand	post bloat	stomata
beginning bloat	ear	leg	prone	supine
black brown discoloration	eggs	legs	prosthetic device	surgical implant
black discoloration	exposed bone	lividity	purge	surgical item
black gray discoloration	eye	livor mortis	raccoon print	tag
black plastic	face	maggot	recessed eye	tattoo
black tarp	flies	maggot eggs	recognizable face	torso
bloat	fly	maggot foam	remove	trauma
body	foam	maggot mass	remove?	trunk
bone	foot	maggots	right foot	waterlogged skin
bone exposure	fresh skin	marbling	right hand	whole body
brick	genitals	medical device	sampling site	worm
zip tie	yellow jacket			

- How many subjects are in the dataset?
- How many days each subject remains in the facility?
- How many images are taken for each subject?
- What is the size and aspect ratio of these images?
- What are some common categories of images in the data?
- What are some outliers and less desired images?

Some of this information might be known by the owners of the data. However, if not, they can also be collected through automation scripts. In the following, two examples are provided to illustrate finding the answers to some of these questions using simple bash scripts.

Exploring the data, we learned its directory structure (Figure 3.2). Furthermore, We realized that for each donor, there might exist three categories of images: “Pickup”, “Intake”, and “Daily photos” representing the images taken from the donors while picked up from the location of death, when they were in a morgue, and when they were in the facility, respectively.

Considering the following directory structure for the human decomposition data and the fact that the files are stored in JPEG (.JPG) format, to find out how many donors exist in the dataset, we first generated a text file with all image names via a bash command.

```
find . -iname ‘‘*JPG’’ > all-images
```

We then used the following command to learn that there are 627 subjects/donors in the dataset.

```
cat all-images | cut -d ‘/’ -f 2| sort -u | wc -l
```

We used the following command and similar ones to learn if all donors have all 3 categories. From the 627 donors, 612, 302, and 492 donors have “Intake”, “Pickup”, and “Daily” photos respectively. Some donors have all three categories, and some have fewer. Overall, bash scripts proved to be a great method for collecting general statistics on the data at the file level.

```
cat all-images | grep ‘‘keyword’’| cut -d ’/’ -f 2| sort -u | wc -l
```

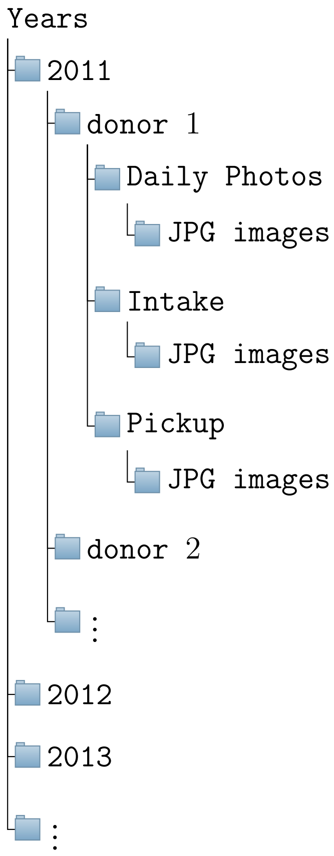


Figure 3.2: File structure of the human decomposition dataset

“Daily photos” are the images that are used to study the human decomposition process and are used in this dissertation. Therefore, only the 492 donors with “Daily photos” are included in our work.

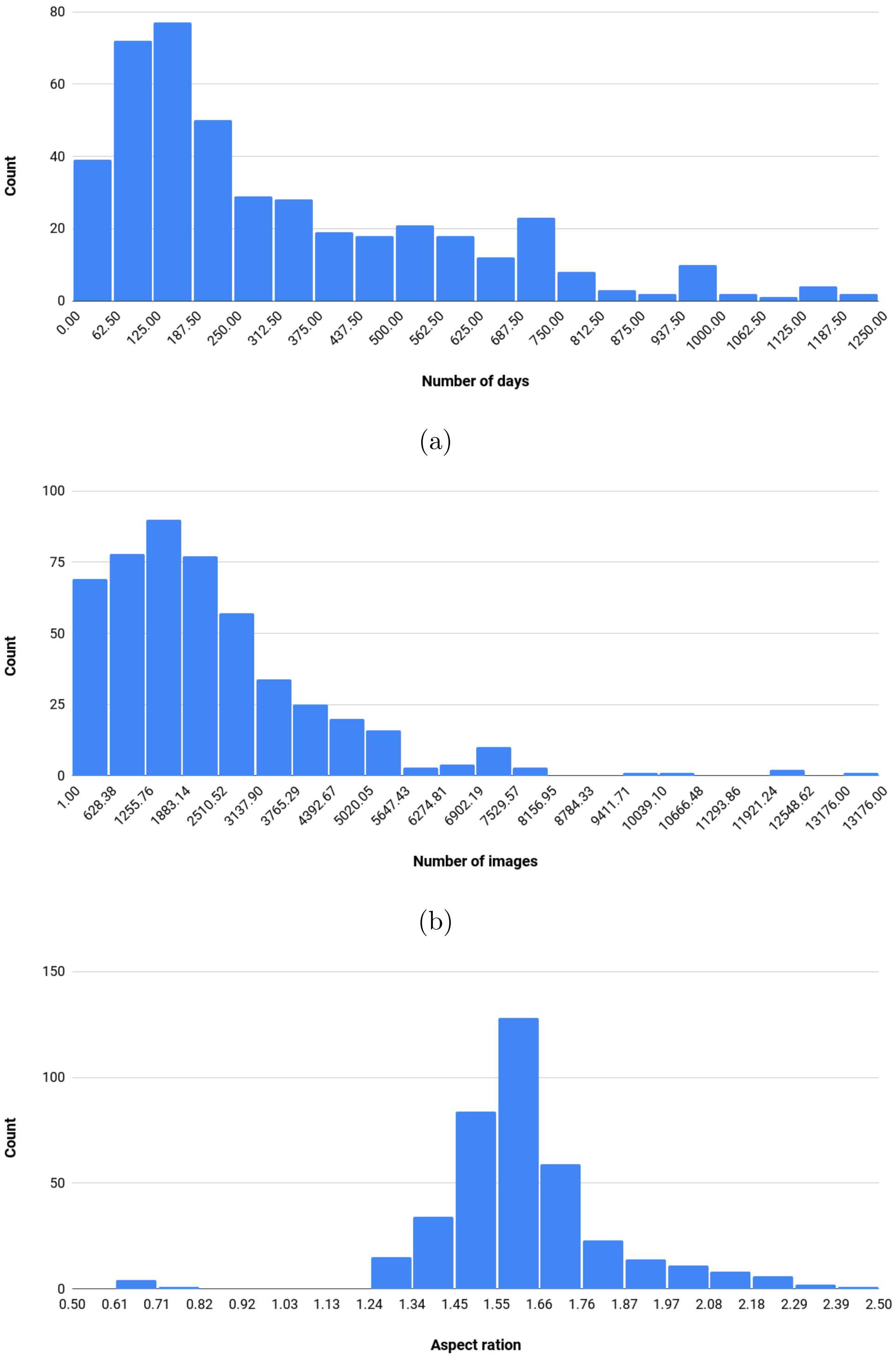
Furthermore, we examined the number of days between the first and the last image for each donor, the number of images for each donor, and the aspect ratio of the width to the height of the images. Histograms 3.3a, 3.3b show the number of days each donor remains in the facility and the number of images for each donor. The histograms show that most donors are in the facility for less than a year, and the number of images taken for the majority of the donors is less than 4000 images. In addition, histogram 3.3c shows the aspect ratio of the width to the height of the images. The histogram indicates that the majority of the photos have a landscape orientation.

3.2.1 Visualizations with UMAP

Once the general information about the dataset is known, it is important to gain insight into the images’ content. In cases such as the human decomposition data, where the images are collected by following a protocol, some general information about the content of the images is known. To obtain as well as confirm this information and gain insight into the images’ content, visualizing the data is key.

For visualizing the images, we built on Parallax [95] to visualize the embedding of the images for a few randomly selected donors. We first generated feature embeddings for the images of interest by feeding them to a ResNet model [58] pre-trained on ImageNet [37]. The resulted embeddings were then used as input to Parallax. Our implementation allows us to use UMAP (Uniform Manifold Approximation and Projection) as the dimension reduction technique to project these embeddings to a 2D space. In addition, it allows us to hover over the data points in the 2D space and see their corresponding images. Figures 3.4 and 3.5 are two examples of our data exploration using Parallax.

Exploring these visualizations, we observed two groups of images depicting “plastic” and “stake” that were outliers compared to the rest of the images. These images are taken from the donors when they are covered by a plastic or from their identifiers which is a wooden stake with their IDs written on it and is placed next to each donor. This observation is



(a)

(b)

(c)

Figure 3.3: (a) shows a histogram of the number of days donors remain in the facility. (b) shows a histogram of the number of images taken from the donors. (c) shows a histogram of aspect ratio of the images for the human decomposition data.

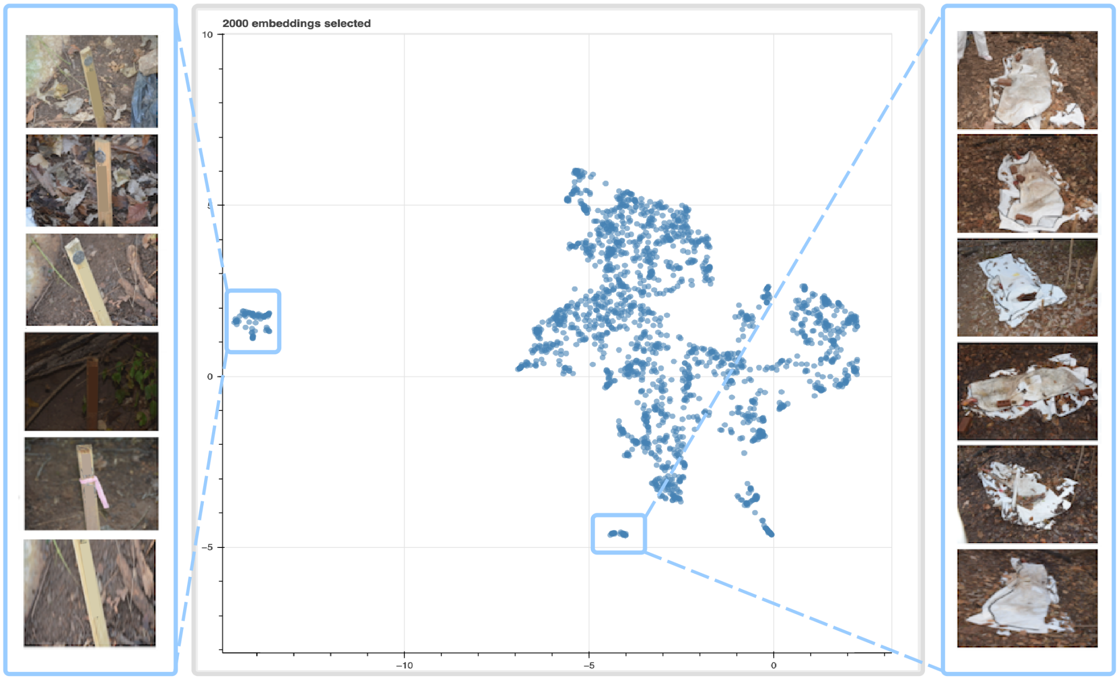


Figure 3.4: Data exploration with the goal of identifying irrelevant content - example 1. The IDs on the stakes are anonymized.

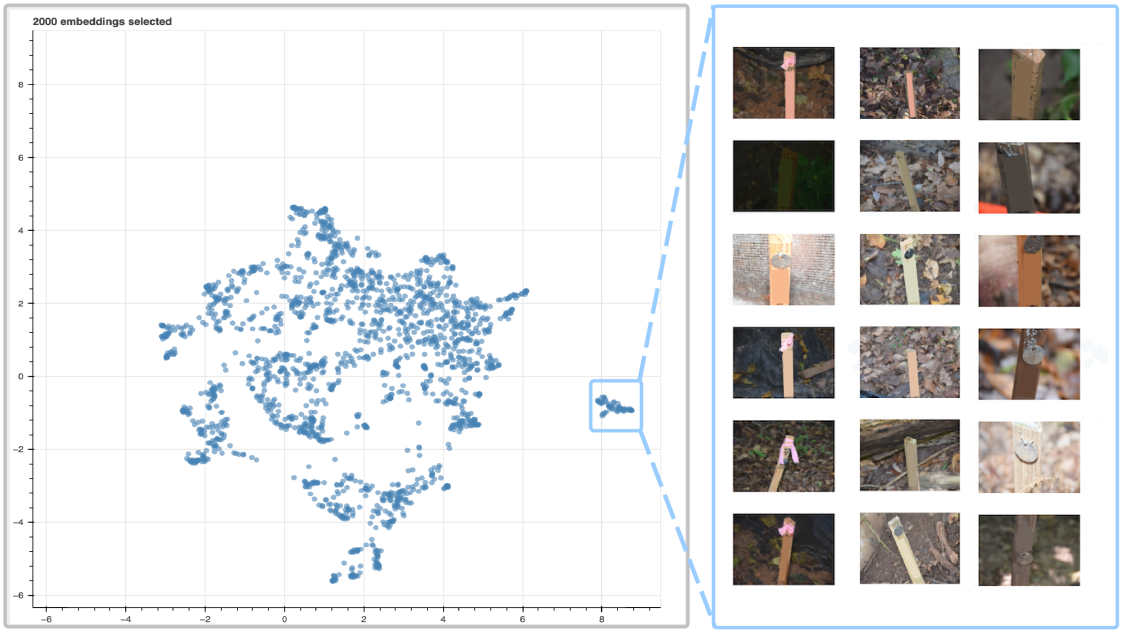


Figure 3.5: Data exploration with the goal of identifying irrelevant content - example 2. The IDs on the stakes are anonymized.

useful because for building models for the decay process, these two sets of images will not bring any value and need to be excluded from analysis.

In addition, we observed when these two groups of images are excluded, the relationship between the rest of the images becomes more clear as is shown in Figure 3.6.

3.3 Data Cleaning

Data quality directly influences the interpretations of the data and the accuracy of the analysis done on it [133]. That is the reason why data scientists spend most of their work hours on data cleaning and improving data quality and data preparation [87].

Data cleaning is an essential step in the curation process and is heavily influenced by the exploration step. The better the data is explored, the better the abnormalities, inconsistencies and errors in the data are detected to be fixed.

Manually collecting an image dataset makes it prone to many inconsistencies in the names and format of the images and duplicate images. Cleaning a large uncurated dataset can be done from a few main aspects as done in [112, 130, 149]: 1) standardizing filenames, 2) deduplication, and 3) removing irrelevant data.

3.3.1 Standardizing Filenames

Having a consistent file format and name convention is very important for working with a dataset. Many interactions with the data, searching for relevant content, or reading files for various purposes are often through automated techniques. Not having a consistent naming will result in not capturing all images, breaking automation, vagueness for the developers who work with the data, or more complicated management of and interactions with the data in general.

In the human decomposition data, each image is originally stored following this format for their names: $UT[xx] - [xx]D\_[mm]\_[dd]\_[yyyy]$ $([X]).JPG$ where x represents a digit. These numbers represent a 2-digit ID for each donor, a 2-digit year of time of placement in the facility, month, day, a 4-digit year (date at which the photo was taken), and a counter

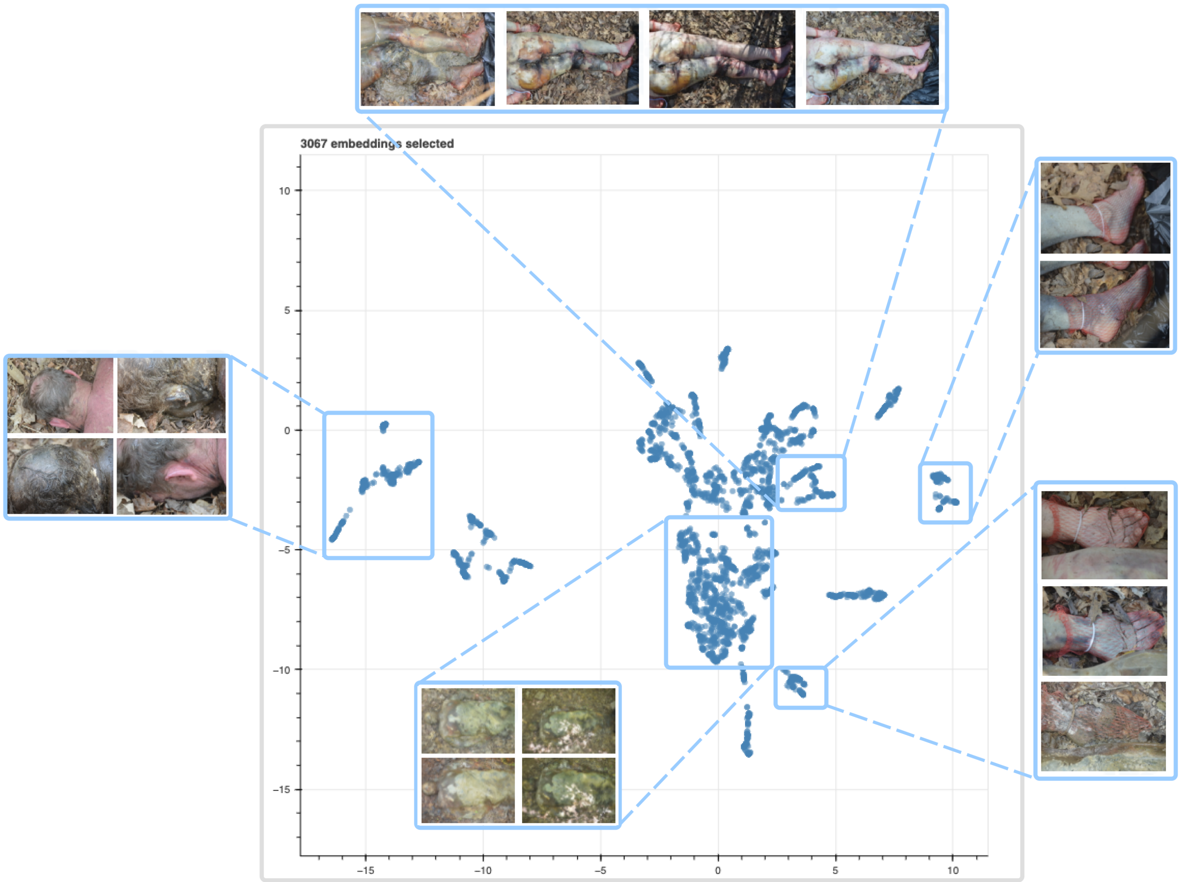


Figure 3.6: Visualizing the feature embeddings of a single donor with UMAP when the ‘plastic’ and ‘stake’ images are excluded.

for the number of images in that date respectively shown with X. An example of such names is $UT28 - 15D\_06\_24\_2015$ (32)$.JPG$.

However, exploring the data, we observed that there are many images whose naming does not follow this pattern. Some photos did not have spaces in their name, some were missing one or both parenthesis, some had a 4-digit year before 'D', and some had a 2-digit year in the date. In addition, there were privacy concerns over the identity of the donors, which prompted us to further anonymize the UT IDs.

In order to have consistent naming and anonymous IDs for the donors, we used a unique 3-digit hex number as an identifier for each donor and removed the year, parenthesis, and space from the names. The new naming follows the pattern of [xxx][x][mmdd].xx.JPG for donor id, number of years in the facility, month, and day at which the photo was taken, respectively. An example of the new naming is $07b00120.14.JPG$.

3.3.2 Deduplication

Duplicate images in a dataset do not provide any additional values but, on the other hand, result in misspending resources as well as causing imbalanced sub samples of the data, consequently hindering machine leaning techniques and statistical analysis. Manually identifying these images and removing them from a large image collection is not feasible. This process, even when automated, can have a long run-time for a large image dataset.

For the human decomposition data, we used image hashing to identify the duplicate images. Image hashing [16] is used to map each image to a hash value. Unlike other hashing algorithms like MD5 and SHA-1, where a slight difference results in a drastic difference in the hash value, image hashing provides similar hashes for images that are similar to each other. Therefore, image hashing can be used to identify highly similar and duplicate photos.

There are various types of hashing, namely average hashing, perceptual hashing, difference hashing, wavelet hashing, HSV color hashing, and crop-resistant hashing [16]. In this work, we used average hashing. We used a script to generate hash values for the entire dataset and to count each unique hash value occurrence. Our script took 5 days to run for 989838 images and detected 63468 images that had at least one extra copy in the dataset.

In addition, in the process of generating the hashes, each image needs to be read first, and if any of the files is corrupted, it will be detected in this process and can be excluded from the dataset.

3.3.3 Removing Irrelevant Content

In addition to the duplicate images, the dataset might include content that does not bring additional value if included in data analysis, but in fact, can even hurt the performance of those analyses. Two examples of such cases in the human decomposition data are images taken from the subjects when they are covered with a plastic bag and the images from the wooden stake that has the ID information of the subjects. These images do not provide any additional information regarding the decay process and can be excluded from the models and analysis done to help study decay. In addition, due to privacy concerns, the photos with stakes were excluded to keep the identity of the subjects anonymous.

To remove these two types of images from the dataset, we trained a simple image classifier on a small set of labeled images obtained from domain experts. The classifier was taught to distinguish between images of plastic, images of stakes, and cadavers. We then used the classifier to classify the images and removed those classified as stake or plastic.

3.4 Data Organization (SChISM)

The key point of clustering is to segment a large collection of observations into a smaller set of groups of similar observations, which helps understand large datasets and organize them. Traditionally, clustering methods group images based on the similarity of features extracted from them. With adequate feature representations, image clustering methods have achieved good results [35, 50, 56, 82] on popular image datasets such as ImageNet, MNIST, COIL100, and VOC2007 [37, 38, 116, 43]. Guérin et al. [56] used pre-trained CNNs on common datasets such as ImageNet to map images to feature representations and then clustered them. Other unsupervised frameworks introduce clustering losses to jointly learn ConvNet features and image clusters in an end-to-end manner [8, 41, 79, 141, 145, 20].

However, evolving images with conceptual likeness with similarity declining over time are not uncommon, yet such data confounds clustering approaches that rely on measures of image similarity as the early stages of the same conceptual object may bear no visual resemblance to the late stages. Supervised techniques might fare better in such situations, but the creation of the labels may have prohibitive costs exacerbated by the inability to do crowdsourcing when domain experts (as in our case forensic anthropologists) may be scarce. In addition, we are not aware of any unsupervised work that clusters image datasets with evolving content based on their semantic similarity.

In the case of human decomposition, a hand appears very different in the fresh stage compared to when it is decayed. Throughout the decay process, bodies undergo many changes that result in different appearance of images depicting the same area of the body. That makes the automatic detection of images depicting the same body part throughout their evolution, for studying their decay process, a difficult machine learning task. Figure 3.7 shows two class examples, foot and hand of this dataset and illustrates the drastic changes resulting from the evolution and decay.

In this dissertation, we introduce a technique for clustering evolving images in the context of human decomposition data. Specifically, our goal is to jointly cluster body parts and decomposition stages within subjects and to trace them through their decay process, which spans from "fresh" to "skeletal". Such an unsupervised approach, if successful, would reduce the manual labeling effort required to extract domain specific features needed for key forensic tasks such as time of death estimation and, more generally, human decomposition research and analysis [100, 101].

Our approach to address the challenge introduced by the evolution of the content of these images is to use a sliding window technique inspired by data stream clustering [5] along with feature representations extracted from pre-trained CNNs [56, 119]. First, we create small sequences, that we call *snippets*, of similar images by maximizing similarities within a sliding window and then stitching these snippets to effectively capture the evolution of the objects based on overlaps and a dynamic inclusion criterion. This stitching results in sequences

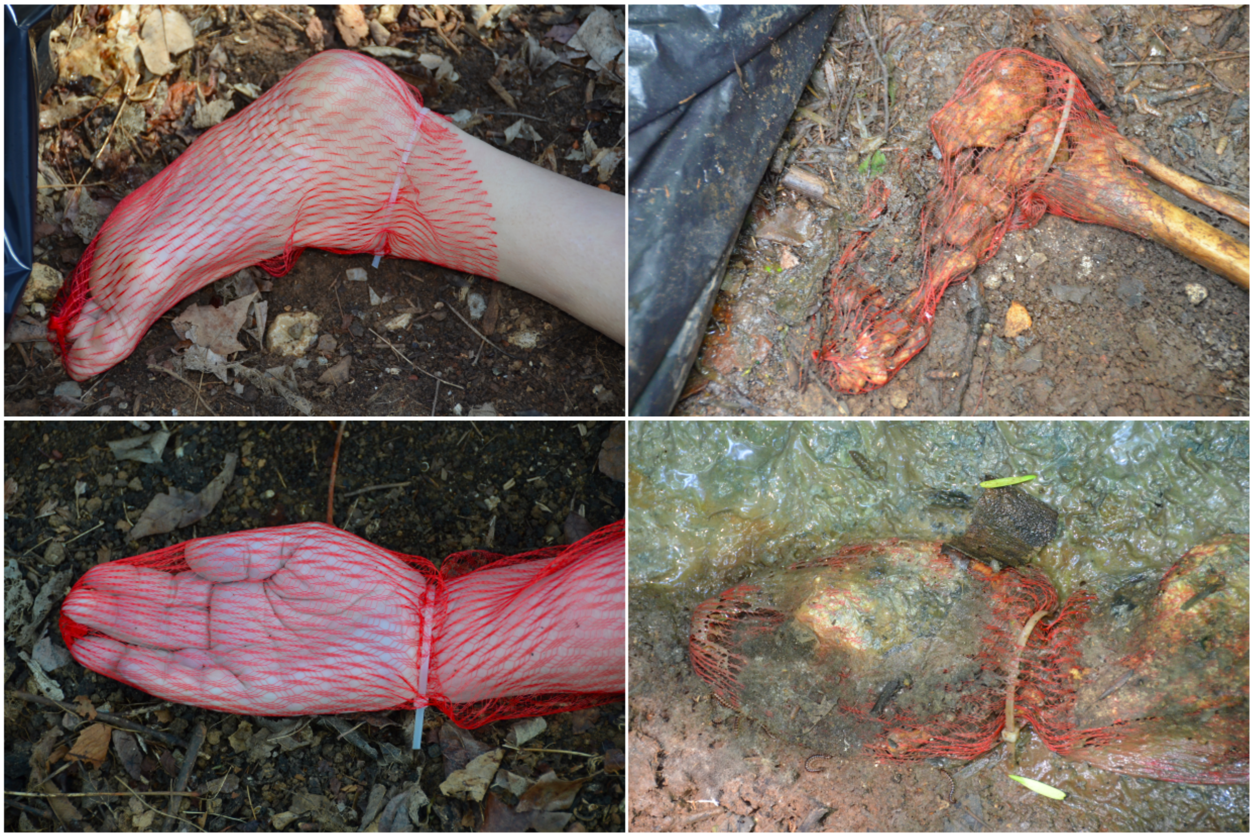


Figure 3.7: Image examples of two classes, foot and hand, are shown in early (left) and late (right) decomposition stages

where images of the same sequence represent the same object (body part) and capture all stages of decomposition from fresh to skeletal. These sequences are essentially clusters of images with a temporal attribute. We refer to our method SChISM as Semantic Clustering via Image Sequence Merging [98] [1] [2].

To evaluate our method, since to our knowledge there is no other work tackling the same or closely similar problem, we compare SChISM with two works that have reported to outperform other unsupervised clustering methods. First is a pre-trained CNN-based image clustering technique [56] which does not involve any model training. The second technique is DeepCluster [20] which is the state of the art clustering based on unsupervised visual representation learning. We use the general clustering metric, purity [89] for comparison. We also introduce new goodness-of-fit metrics that are more suitable for datasets with evolving content in Section 3.4.2.

In addition to the above comparisons, we also test SChISM on the MORPH dataset [111] introduced in Section 1.3.2 that contains mugshots of different individuals over time (ranging from a few days to a few years apart). This dataset has similar characteristics to the decomposition dataset, where the goal is to trace and recognize faces as they age.

In the following, we first provide details about the architecture of SChISM and then our evaluation and results.

3.4.1 Architecture Design

The goal of our method is to group large collections of unlabeled but semantically-related images. Even though semantically-related, the images may have distinct appearances due to, for example, evolution over time or representing different context. We further assume that we have partial metadata such as the timestep for each image and some implicit constraints such as the presence of images for some semantic concepts at each timestep.

First, we want to emphasize that such situations are not uncommon in cases where the image collection is subject to certain rules or protocol. Second, we would like to leverage the

[1] Link to the repository: https://github.com/saramsv/SChISM

[2] Mousavi, Sara, et al. "SChISM: Semantic Clustering via Image Sequence Merging for Images of Human Decomposition." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021

semantic relationships and implicit constraints to produce a fully unsupervised clustering algorithm that is capable of using these constraints to produce more accurate semantic clusters. Conceptually, our approach can be described as a penalized optimization problem where we optimize for the visual similarity of images within groups but penalize for the violation of the implicit constraints.

In the case of the human decomposition dataset, the context metadata includes the subject and timestep. Each body part represents a distinct semantic concept. Decomposition makes the images of the same body part look different over time and our implicit constraints are defined by the data collection protocol that requires images of each body part at each observation. We do not, however, have body-part labels in our metadata.

While image similarity can easily group each body part into a single cluster for a specific short duration in time where the state of decomposition is the same, similarity breaks down over longer periods. To address this, we penalize sequences representing short timespans or sequences with long time gaps. In essence, we aim to minimize the loss function of the following kind:

$$loss = \frac{1}{|S|}(\sum_{k=1}^{|S|}(1-(\frac{1}{\begin{pmatrix}|S_k|\\2\end{pmatrix}}\sum_{j=1}^{|S_k|}\sum_{i=1,i>j}^{|S_k|} Sim(x_i,x_j))) + \frac{1}{|S_k|}) \tag{3.1}$$

where S denotes the generated sequences, S_k is the k-th sequence, $|S|$ is the total number of sequences and $|S_k|$ is the total number of images in the k-th sequences, and x_i is the i-th image in a given sequence. The entire parameter search space is extremely large as there are 9 different classes for which images have to be clustered per timestep and then sequenced through 50 timesteps on average for each subject, for a total of 500 subjects (one million images). To make this search feasible, we break it into two stages: 1) grouping of images into short sequences (snippets) that maximizes the similarity within that snippet, and 2) stitching the snippets into longer sequences (final clusters) with minimal gaps. We use the term snippets to refer to partially constructed semantic clusters and use the term stitching to denote the process of iteratively enlarging these incomplete semantic clusters via semantic similarity. We use this terminology to highlight the difference between the merging of images into clusters based on image similarity and based on semantic similarity as specified in the loss function above. The two terms were chosen to indicate that semantic similarity concerns the challenge of constructing contiguous time sequences of the semantic concepts.

Our method consists of three main steps, shown in Figure 3.8. First, we generate feature representations from input images using a CNN model pre-trained on ImageNet [37]. Second, we group images into snippets, and finally, we stitch similar snippets together to form long sequences. In the following, the details of each step are provided.

Producing Image Features

In evolving image data such as images of human decomposition, each timestep has a group of images, subsets of which belong to various classes. We denote these images by $\text{img}_{ti} \in N$, where $t \in \{1, 2, \cdots, T\}$, $i \in \{1, 2, \cdots, m\}$ for T timesteps and m images per timestep, and N represents the set of all images. Note that the number of classes in each timestep is less than or equal to m since multiple images in each timestep may belong to the same class.

The first step in our method is to extract from each image feature representations that are used to capture image characteristics and serve as a basis for comparisons. For this, we feed the images into a pre-trained CNN model excluding the last fully-connected and softmax layers. We used ResNet50 [58]. Other CNNs such as Inception [128] may also be used. The resulting features are then stored as feature vectors for each input image. In the case of using ResNet, each vector has a length of 2048. We denote the feature representation for image i from timestep t as R_{ti}. Inspired by Caron et al. [20], we reduce the length of these representations to 256 using Principle Component Analysis [138] to improve the overall run-time of our method.

Window-based Sequencing

In the decomposition data, typically there are one or more images representing the same class at each timestep. Additionally, the same decomposition stage may correspond to multiple consecutive timesteps and the time span may vary for different body parts. For example, the first and last 3 timesteps might represent fresh and skeletal stages respectively. As a result, there is often more image level similarity within the images of the same decomposition stage

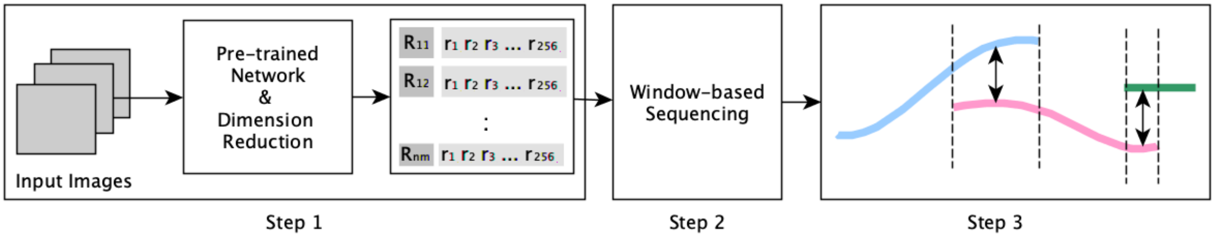


Figure 3.8: The overall architecture of SChISM is shown. Step 1: Input images are mapped to feature vectors. Step 2: The neighboring feature vectors, are then compared to each other and snippets of similar images are created. We use a sliding window on the timesteps to find the neighboring feature vectors (shown in Figure 3.9). Step 3: Snippets are then stitched together to form longer sequences that capture the entire evolution of objects.

rather than across stages, which makes it a challenge to find and trace all stages of decay for a specific class without confusing it with other classes.

SChISM leverages the fact that images from neighboring timesteps are more similar to one another in terms of their local features than those from more distant timesteps. We use this constraint in our data to reduce the size of our search space and create sequences of similar images from the same class over time. Given a series of consecutive timesteps T, we define a sliding window W. Each image representation R_{ti} in W is compared to all of the images within the window except for the images of its corresponding timestep (Figure 3.9). If the similarity between R_{ti} and another image $R_{t'j}$ where $t' \in W$ and $t' \neq t$, is greater than a threshold, $R_{t'j}$ is added to the short sequence (snippet) that R_{ti} is a member of. If such a snippet does not exist, it is created with the two images included.

When image classes change over time, the level of similarity between images of the same classes may vary depending on the timesteps and the state of the decomposition. Therefore, if a constant threshold is used to decide if an image should or should not be added to a snippet, classes may be miss-linked. We use a dynamic threshold to overcome the varying similarities. The threshold is set to

$$max(\alpha \times Sim_{max}(R_{ti}, R_{t'}), \beta) \tag{3.2}$$

where α and β are constant values. This process results in a series of snippets in which images that have the most similarity throughout time are connected, essentially grouping an image class along with its evolution. For image comparison, any two vectors, R_{ti} and $R_{t'j}$, are compared using cosine similarity as

$$Similarity(R_{ti}, R_{t'j}) = \frac{R_{ti}.R_{t'j}}{\|R_{ti}\| \,.\, \|R_{t'j}\|}. \tag{3.3}$$

Stitching Short Sequences

Due to the possibility of having multiple images for each class at any given timestep, the resulting snippets may have image or time overlaps. To maximize the length of final sequences for each class, we use three levels of stitching.

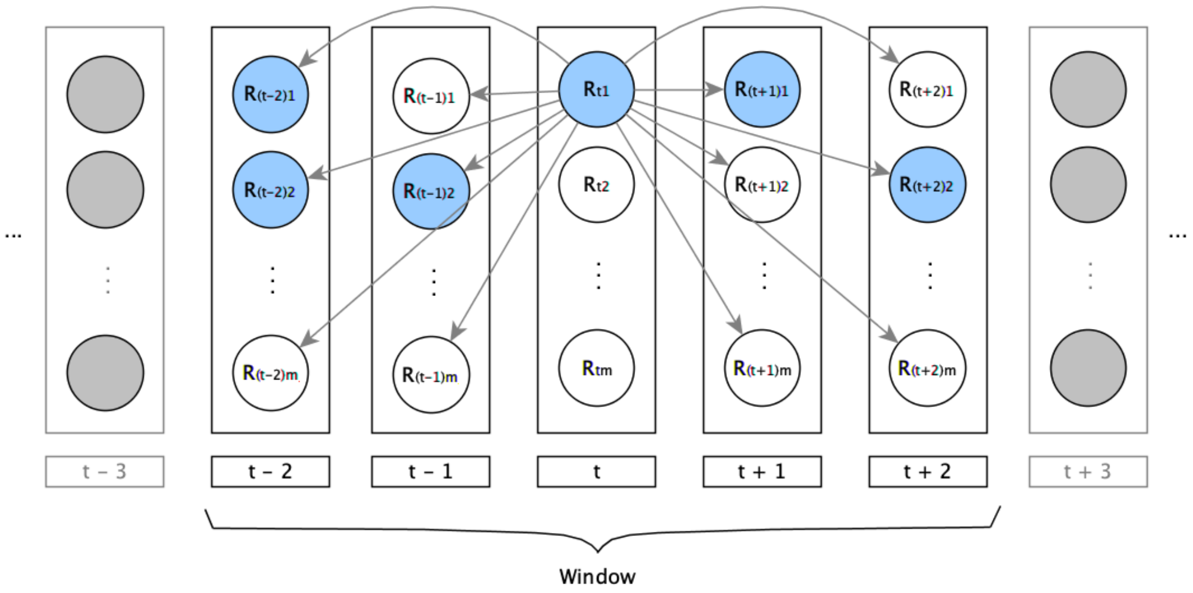


Figure 3.9: Each image in timestep t is compared to all images in other timesteps within the boundary of the sliding window W. After comparing all images of t, the sliding window is moved forward by one timestep.

First, we stitch snippets that share one or more images. We call this image-overlap stitching.

Second, we stitch snippets with temporal overlaps provided that their similarity is above a constant threshold. To do so, we sort the images in each snippet based on their corresponding timesteps and find snippets that have time overlaps. For each pair of snippets, we set the one that starts with images from earlier timesteps as the first and the other one as the second. The tail of the first snippet is compared with the head of the second. The comparison is done by measuring the cosine similarity between every two image pair in the time overlap. If the average similarity of the overlap is greater than η, the two snippets are stitched together to form a longer sequence. Note that $\eta < \beta$. While this gives a second chance to stitch snippets of the same class that have not yet been grouped together using the moving window, reducing β does not have the same effect. That is because β considers inter-image similarity, while η considers inter-snippet similarity.

Third, we attempt at stitching snippets with the goal of filling the gaps that we do not expect them to have based on implicit constraints on the data, namely knowledge of the possible timesteps that exist in the data for a particular subject. To this end, each snippet is only compared against snippets of images that have time intersection with the missing timesteps in the current snippet. The comparison is done using cosine similarity between the average feature vectors of the two compared snippets. The snippets with the highest similarity are then stitched together.

3.4.2 Evaluation and Experimental Results

In order to evaluate SChISM, we used images depicting human decomposition as our primary dataset as well as the MORPH dataset. Both datasets are described in Section 1.3. In the following, details about evaluation metrics, the cluster evaluation process and its interface, and the results are provided.

Metrics

The goal of our method is to group images from the same body parts together, even-though they may look different due to decay, such that the inclusion of images of the same body part from all possible consecutive timesteps in the same cluster is maximized while the gaps in each cluster are minimized.

To evaluate the clusters produced by SChISM, we use the purity metric [89] which is defined as the ratio of correctly clustered images with respect to the dominant class in clusters, to the size of the clusters as in the following:

$$Purity_{class} = \frac{\sum_{c=1}^{\#clusters} C_c - M_c}{\sum_{c=1}^{\#clusters} C_c} \tag{3.4}$$

where C is the set of clusters for the given class and M is the number of misclustered images. However, because purity increases with an increase in the number of clusters, it cannot assess the quality of clusters with evolving contents alone. Therefore, we also define three new metrics namely 1) *gap*, 2) number of *essential clusters*, and 3) *inclusion*.

Gap is defined as the number of missing images corresponding to consecutive timesteps in each cluster. For example, if a subject is photographed over 10 sessions, the corresponding timesteps are $\{t_1, t_2, \cdots, t_{10}\}$. For a given body part of the same subject if it is photographed in every session, the ideal scenario for the resulting cluster should include images corresponding to all timesteps. If the timesteps captured in the cluster are $\{0, 0, 1, 1, 0, 1, 1, 1, 1, 0\}$ (1 if there is an image corresponding to the timestep in the cluster, 0 otherwise), the gap sizes are $\{2, 1, 1\}$ (the total gap for the cluster is 4) and the size of the snippets are $\{2, 4\}$ (the total length of the sequence is 6). The smaller the total gap values are the better the clusters are, in terms of tracing decomposition.

Clustering may result in multiple clusters for each class. To identify the most relevant clusters for each class, we define the **essential-cluster** metric, which is the number of non-subset clusters produced for a given class. As an example for an essential cluster, if clusters C_1 and C_2 include images for the same class corresponding to timesteps $\{t_1, t_2, t_4, t_5, t_6\}$ and $\{t_5, t_6\}$ respectively, C_1 is considered as an essential cluster rather than C_2, since C_2 is a subset of C_1. The lower the number of essential clusters for each body part (class) the better the performance of the clustering method is. In another word, the ideal scenario for each body part is to have one single cluster that includes images for all timesteps. Note that this evaluation metric can only be calculated with known class labels.

Inclusion is defined as the total number of timesteps included in the essential clusters for each body part. In other words, it indicates how many timesteps for each class are captured within essential clusters.

Cluster Evaluation

The human decomposition dataset is not labeled. In order to evaluate the performance of SChISM, we labeled a subset of the dataset to be used as test data. We developed a web interface to facilitate the labeling process and visual evaluation (described in more details in Section 4.2.1). Using the interface, one can label the entire cluster with a class name if the entire cluster represents one class. If misclustered images exist in a cluster (images depicting a different class than the dominant one), the user can first assign a correct label to those images and then label the remaining images at once based on the dominant class. We used this interface to facilitate and speed up the manual labeling of our test data which includes $34,476$ images corresponding to 10 randomly selected subjects from the human decomposition dataset.

Results

In order to test our method, we used the $34,476$ labeled images mentioned above. We compared the resulting clusters from SChISM with those obtained from a naive baseline as well as the following methods from [56] and [20]: a) pre-trained CNN-based image clustering, b) pre-trained DeepCluster, and c) trained DeepCluster, on the $34,476$ selected images using metrics introduced in the Metrics section as well as the purity metric.

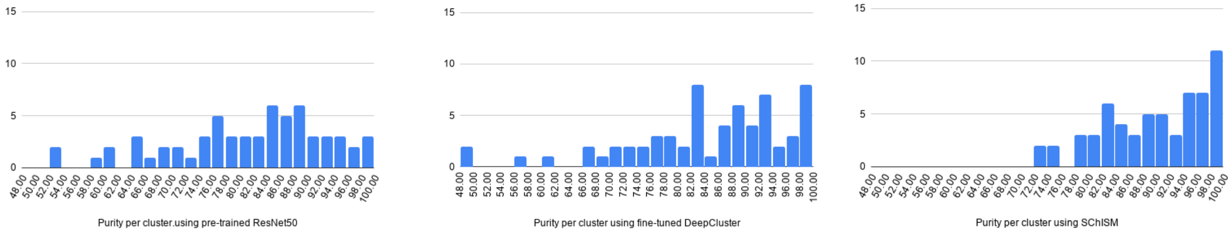
The **naive baseline** simply uses a non-trained CNN to map the images to feature representations and then clusters them using KMeans. **Pre-trained CNN-based image clustering** [56] is a trained version of the naive baseline method where the network is pre-trained on a common dataset such as ImageNet and then feature representations obtained from applying the network on our test data is fed to KMeans for clustering. We used

ResNet50 as the CNN for both approaches for the sake of comparison with SChISM. The **pre-trained DeepCluster** method [20] consists of clustering our test data using DeepCluster pre-trained on ImageNet. Finally, we compared our results with **trained DeepCluster** which is trained on our data. Note that training DeepCluster is an unsupervised process. We did not compare against a supervised method such as training or fine-tuning a CNN on our data since our method is unsupervised and we do not have training data. The purity histograms for pre-trained CNN-based image clustering, trained DeepCluster, and SChISM are shown in Figure 3.10.

Tables 3.2 and 3.3 show the average purity for each class (body part) across all ten subjects, as well as additional statistics. The number of clusters used for all methods was set equal to the number of clusters obtained from SChISM, which was, on average, 72 clusters for each subject. The hyper-parameters used in our implementation of SChISM were $\alpha = 0.99, \beta = 0.7, \eta = 0.65$ and $W = 4$. Our analysis on different values for α, β, and η show that the higher the values, the more restrictive the inclusion criterion becomes and therefore results in a larger number of sequences and clusters and higher purity values. Lower values, however, result in loosening the criterion, smaller number of sequences and clusters, and lower purity values. Higher values for W increases the chance of images from distant days being compared with each other and therefore increases computation.

Note that while increasing SChISM's hyper-parameter values results in more clusters and consequently higher purity, higher number of clusters has the same effect in other unsupervised clustering methods as well. However, Figure 3.10 shows that SChISM results in higher purity for the same number of clusters as used by other methods.

In addition, we further evaluated our method using the number of essential clusters, gaps, and the inclusion metrics introduced in Section 3.4.2. The results along with a visualized example of the clusters generated for one subject are shown in Figure 3.11. As Figures 3.11a and 3.11b indicate, images from similar number of timesteps are captured using smaller number of essential clusters in SChISM compared to that of pre-trained CNN-based image clustering and the trained DeepCluster. This indicates that sequences are generally longer in SChISM than in the other methods. In addition, we noticed that some classes can have zero clusters in the other methods. For example, for class 'arm', the other methods did not



(a) (b) (c)

Figure 3.10: pre-trained CNN-based image clustering, trained DeepCluster and SChISM. The majority of the clusters obtained using SChISM have purities higher than 84% for the same number of clusters.

Table 3.2: Per body part purity for all clusters with at least 5 images are provided. These values are averaged for the 10 selected subjects. The number of clusters for all methods were set to the same value obtained from SChISM for a fair comparison.

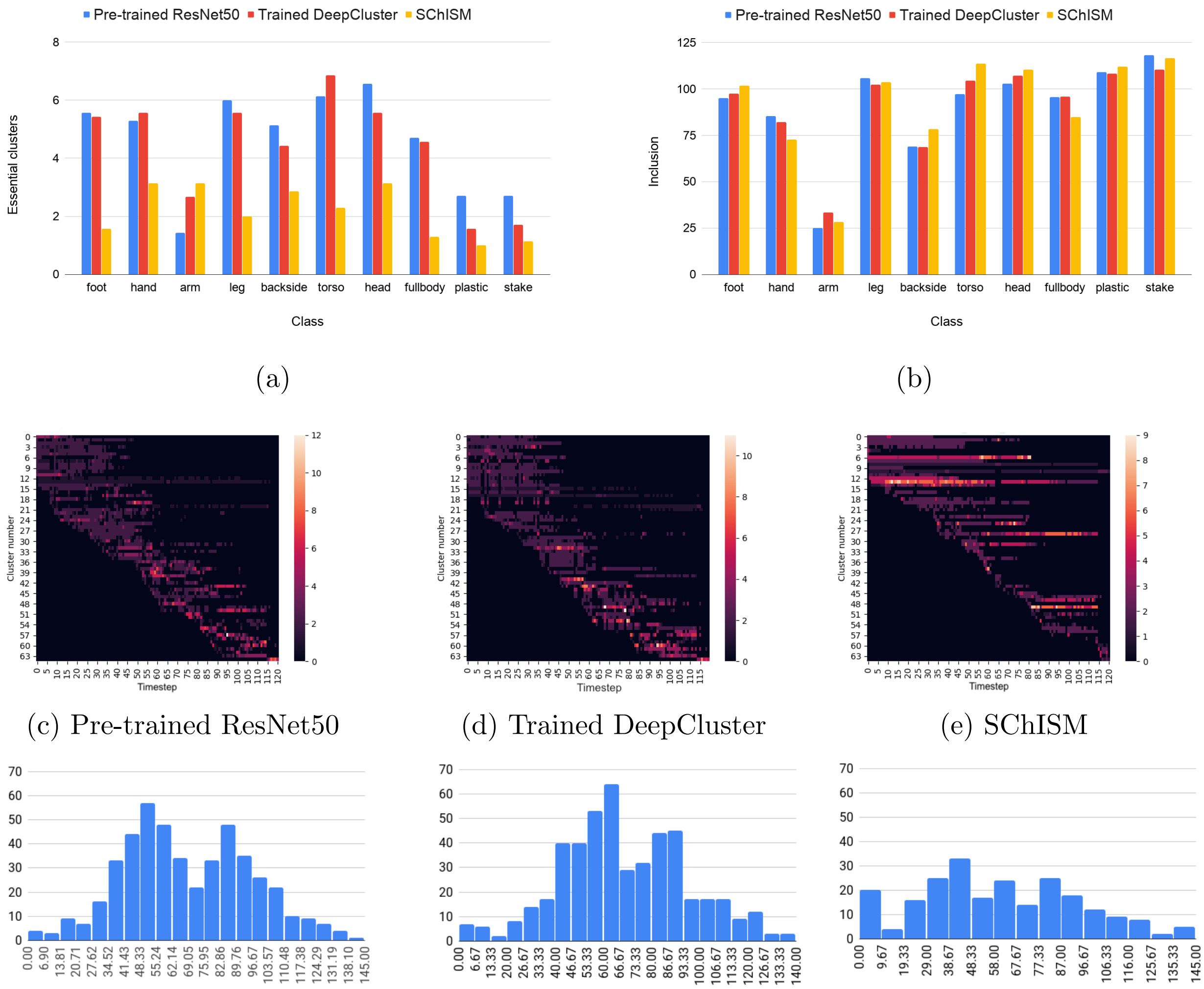
Human Decomposition										
Method	**Average purity per class (%)**									
	Stake	**Foot**	**Head**	**Full**	**Plastic**	**Torso**	**Arm**	**Leg**	**Back**	**Hand**
Naive baseline	49.71	79.78	79.57	76.49	68.01	65.78	60.09	80.18	56.86	76.11
Pre-trained CNN	99.27	88.004	92.29	75.02	94.22	78.33	77.16	81.77	69.03	83.31
Pre-trained DeepCluster	96.57	81.34	91.69	77.29	96.11	78.53	78.05	78.14	62.41	81.93
Trained DeepCluster	96.85	90.73	94.07	84.97	90.89	79.91	87.92	82.87	69.93	86.76
SChISM	99.14	95.42	96.21	88.03	96.96	85.88	88.23	87.32	84.58	95.41

Table 3.3: Mean average purity, standard deviation, median and min for all clusters with at least 5 images are provided. These values are averaged for the 10 selected subjects. The number of clusters for all methods were set to the same value obtained from SChISM for a fair comparison.

Human Decomposition				
Method	**Avg Purity**	**Standard Deviation**	**Median**	**Minimum**
Naive baseline	72.71	18.78	25.73	74.02
Pre-trained CNN	83.50	11.78	85.13	52.49
Pre-trained DeepCluster	81.24	14.89	83.12	31.82
Trained DeepCluster	85.99	11.37	88.52	49.72
SChISM	**92.30**	**7.27**	**91.93**	**72.22**

produce any cluster for some of the subjects. Such scenarios do not happen in SChISM due to its temporal matching. Furthermore, histograms on gaps for all clusters with more than 5 members generated using pre-trained CNN-based image clustering, trained DeepCluster, and SChISM are shown in Figures 3.11f, 3.11g, and 3.11h and show that clusters generated using SChISM have minimum gaps compared to the other methods. We consider clusters with less than 5 members as outliers with images that do not capture body parts and could not be stitched to any of the larger sequences. We did not include the naive baseline and the pre-trained DeepCluster since pre-trained CNN-based image clustering and trained DeepCluster are the more accurate versions of the two respectively. Purity values for all approaches, however, are shown in Table 3.3.

Finally, we clustered mugshot images using SChISM to assess the performance of our method on a different dataset with temporal evolving content. We selected subjects with at least 5 timesteps which resulted in 417 images from 11 subjects. We then used SChISM to group images based on subjects irrespective of their age (date of the mugshot). The result and statistics on the clusters are shown in Table 3.4. Furthermore, Figure 3.12 shows two clusters generated using SChISM for ‘foot’ and a subject at ages 41, 50, 51 and 52 from the decomposition and the MORPH datasets.



(a)

(b)

(c) Pre-trained ResNet50

(d) Trained DeepCluster

(e) SChISM

(f) Pre-trained ResNet50

(g) Trained DeepCluster

(h) SChISM

Figure 3.11: (a) and (b) compare pre-trained CNN-based image clustering, trained DeepCluster, and SChISM through the number of essential clusters and number of timesteps captured for each body part (inclusion). The plots show that SChISM was able to capture same or higher number of timesteps in smaller number of essential clusters. (c), (d) and (e) compare the clusters generated from the methods for one subject respectively through a visualization. Color indicates the number of images in each timestep and cluster. The plots indicate that SChISM generates clusters with longer sequences and with less gaps in timesteps compared to the other methods. (f), (g), and (h) show gap histograms for clusters and indicate that clusters generated using SChISM have minimum gaps compared to the other methods.

Table 3.4: Statistics on clusters generated for the MORPH dataset using SChISM.

MORPH						
# images	**#Subjects**	**#Clusters**	**Purity**			
			Std.	**Avg.**	**Med.**	**Min.**
417	11	15	0.48	99.87%	100%	98.14%

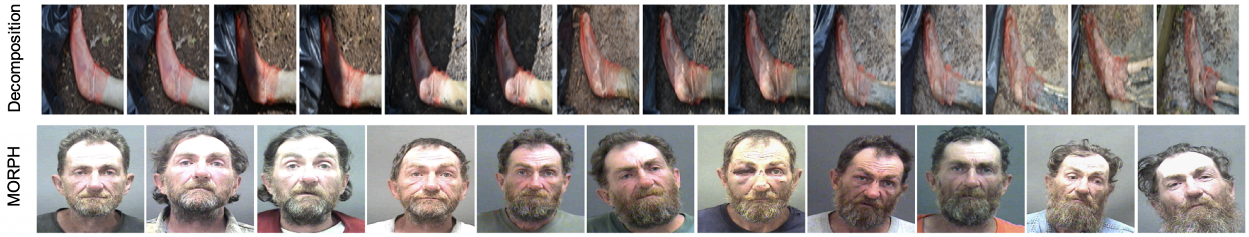


Figure 3.12: Example clusters obtained using SChISM.

Chapter 4

Data Enrichment

We define data enrichment as the process of extending the data with external labels and meaningful information that can facilitate further research, analysis, and queries.

This extension may come in the forms of 1) associating new types of information gathered from external sources to a dataset (e.g., labels obtained from the metadata of the dataset, labels provided by domain experts, or labels obtained from another dataset to the images), 2) extending already-existing metadata or labels to the entire dataset (e.g., extending image-level annotations to pixel-level segmentation or learning from the existing labels for some images and generating labels for the unlabeled ones).

Associating labels obtained from metadata or external datasets to a target dataset can be easily automated. However, in cases that human intervention is needed to obtain these labels, it is costly or even infeasible to do so in a fully manual manner. In our auto-curation framework, we present techniques to assist the manual labeling process and automatically expand on what they provided to the unlabeled portion of the data. An overview of the data enrichment phase is shown in Figure 4.1.

4.1 Automatic labeling using metadata

Automatic labeling can be done using available and obtainable information from the metadata of the dataset or external datasets. Labeling the images with metadata or metadata-inferred information is a straightforward and intuitive task and can provide an

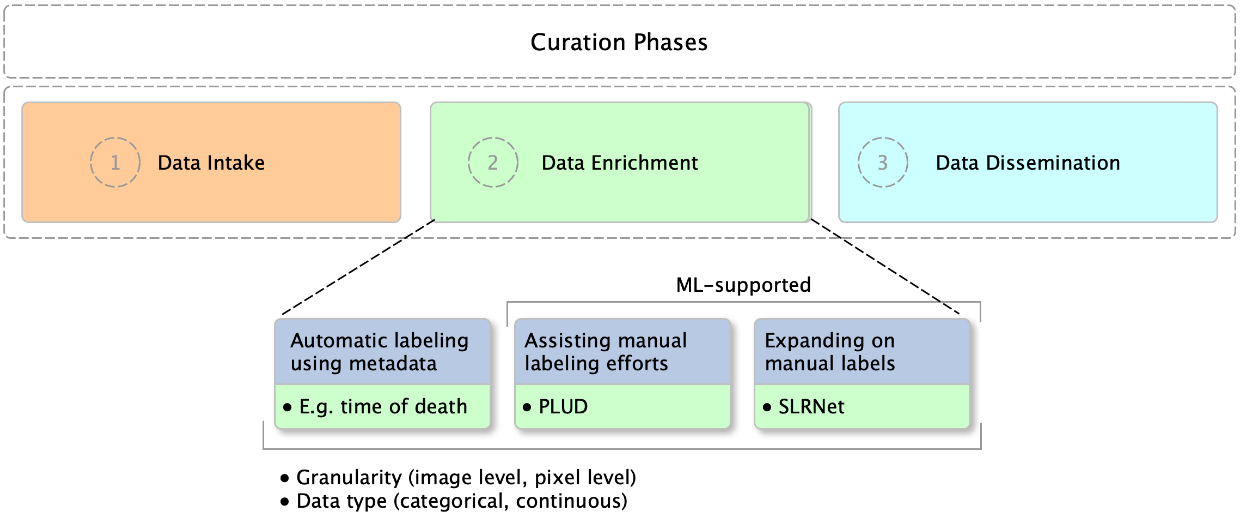

Curation Phases
1
Data Intake
2
Data Enrichment
3
Data Dissemination
ML–supported
Automatic labeling using metadata
• E.g. time of death
Assisting manual labeling efforts
• PLUD
Expanding on manual labels
• SLRNet
• Granularity (image level, pixel level)
• Data type (categorical, continuous)

Figure 4.1: Data enrichment components.

initial structure for the data based on this information. Some examples of such information are date or camera information. In addition, other meaningful labels can be obtained from external datasets to be associated with the target dataset.

In the human decomposition data, we have the donor and the date of each image associated with it. This dataset can be enriched with other information, such as age, sex, weight, the ancestry of the subjects, and weather history prior to the date of the images, without human intervention. Having external demography and weather data, the ID of the subjects, and the date of the images, we can find demographic information and the weather history for the images and properly link them to the images.

4.2 PLUD: A Platform for Labeling Uncurated Data

Although by using a clustering algorithm, an image dataset can be organized into groups of images with similar characteristics, for answering many research questions, it is still required to have explicit labels expressing what each image holds. Manually obtaining such information for a large image collection is not feasible, motivating the urge to automate this process using classification algorithms. However, a good classification model requires a large amount of image-level labeled training data. To address the scarcity of labeled data, we develop an iterative, semi-supervised method to build a high-performance classifier starting from limited available labeled data.

We introduce a Platform for Labeling Uncurated Data called PLUD [100] [1] [2] to support this task. PLUD is a human-machine collaboration-based platform to semi-automate the image-level labeling process and reduce human effort. PLUD is an iterative collaboration between a clustering algorithm, a convolutional neural network (CNN) based classification method, and a human expert to supervise the correctness of the labels.

In the workflow that PLUD provides, users start with a large unlabeled set of images and produce accurate domain-specific models that can be used to label and consequently

[1] Link to the repository: https://github.com/saramsv/PLUD

[2] Mousavi, Sara, et al. "Collaborative learning of semi-supervised clustering and classification for labeling uncurated data." 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020.

clean and structure the collection. That, in essence, unlocks the full potential of the content hidden in these images to solve research and practical problems.

The main components that enable PLUD to achieve its goals are: 1) the use of unsupervised clustering approaches to group images, 2) a user interface that supports rapid labeling of large batches of samples, and 3) iterations that continuously increase the accuracy of the model while reducing the labeling effort of the user.

Specifically, the iterative sequence of clustering, labeling, and classification gradually introduce new unseen data in each iteration. The resulted labels from each iteration are used as input data for a classification model in the next iteration. High confidence results from the classifier are used to boost its learning, while low confidence decisions route images to the clustering and manual labeling component in order to be corrected. Repeating the iterations results in more labeled data and more accuracy for the classifier, and hence less need for manual labeling in consequent iterations.

We used PLUD to classify the human decomposition images and evaluated the performance of our classification on a subset of 5555 randomly selected and manually labeled images. The results show that PLUD is able to classify these images with top-1 and top-3 F-scores of 79.89 and 93.84 respectively.

In the following sections, detailed information about PLUD's architecture, the data used for its evaluation, and our conducted experiments on PLUD are provided.

4.2.1 Architecture Design

Figure 4.2 shows an overview of PLUD. The basic idea of PLUD is to combine unsupervised clustering, and supervised classification methods in an iterative workflow involving a human expert, with a user interface tailored for enabling rapid labeling of a large number of images. In each iteration, the uncertainty of the classifier is used to suggest additional data to be clustered and then manually labeled. Images that are considered similar are presented in large batches to leverage the power of the human visual system to detect outliers. The domain expert produces names/labels for each group based on the dominant class in the cluster. The resulting labeled data are then used for training and fine-tuning the classifier in the next iteration.

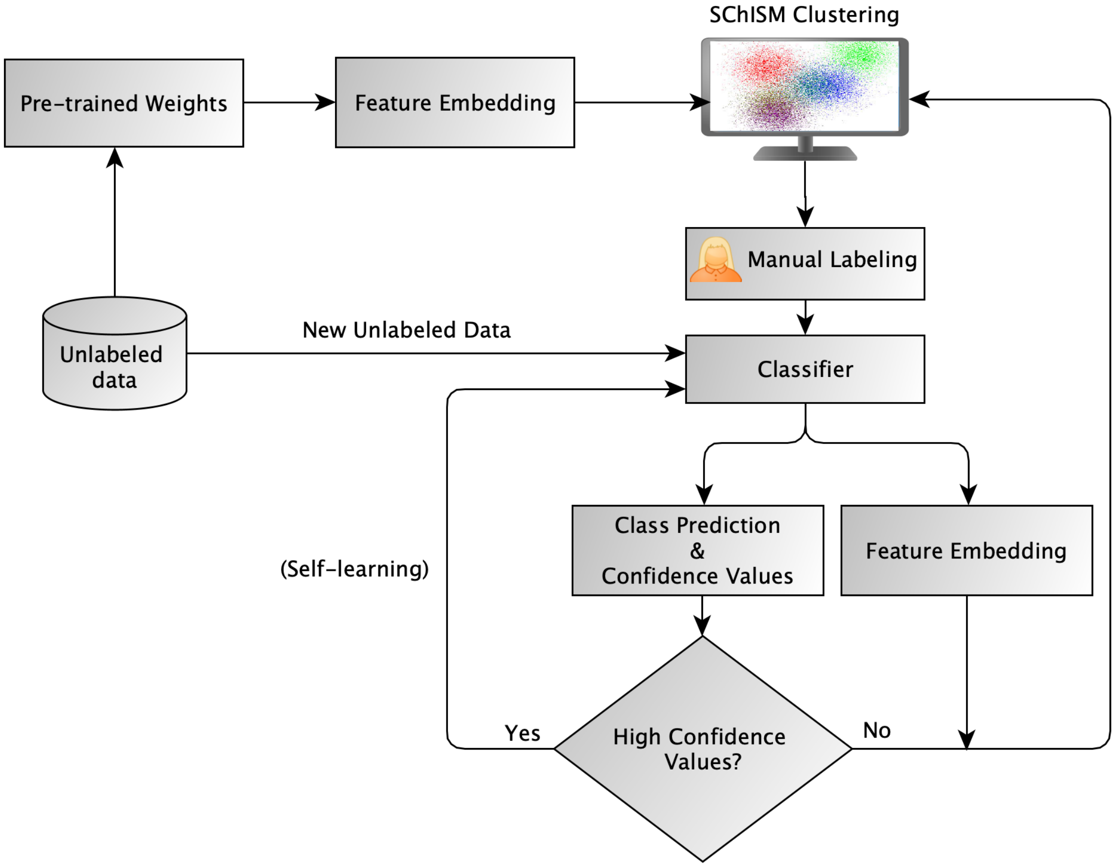


Figure 4.2: Shows the overall architecture of PLUD. Unlabeled data are mapped to numerical feature embeddings and then clustered together. The resulted clusters are displayed to the domain expert for manual labeling through the PLUD interface. The labeled data is used for training a classifier by which, for a new set of unlabeled data, labels and feature embeddings are generated. The iteration repeats for images with low confidence predictions. High confidence predictions are fed back to the classifier to enable self-learning.

In the following, details about data preparation, clustering, the labeling interface, and the classification steps are provided.

Data Preparation

The performance and generalizability of classifiers depend on the quality and the size of their training data. The more diverse and representative the training data is, the more generalizable the models are likely to be. In the case of a temporal dataset, such as images documenting human decomposition in which the subjects' appearance change over time, it is important to sample the training data in such a way that includes the dataset's characteristic. In the case of the human decomposition data, for example, images of all possible decomposition phases should be included in the training data. To ensure the inclusion of changes over time on the human bodies, we randomly selected a small number of subjects and then selected all images taken from them over time, from fresh to decay, rather than randomly selecting images from the entire set which might include more subjects but might also exclude some decomposition phases.

Clustering

In order to assist and speed up the manual labeling process and enable mass labeling, we built a web-based interface that can present the clustered images to the users to be viewed and labeled. To build the clusters, the unlabeled data are mapped to feature embeddings using a convolutional neural network pre-trained on Imagenet [37]. In the first iteration that we had no labeled data, we used a ResNet-based model. We fed in the images to the network and extracted the output of the convolutional layers as the feature embeddings and then clustered them. For subsequent iterations, we used the feature embeddings generated by our trained classifier model. For clustering images, we used SChISM (described in Section 3.4) as our clustering algorithm.

The resulted clusters were displayed to a user for evaluation and cleaning through our web interface. The interface allows users to label individual images or select multiple or all images at once and provide a label for the selected ones. More details about this interface is shown in Figure 4.3.

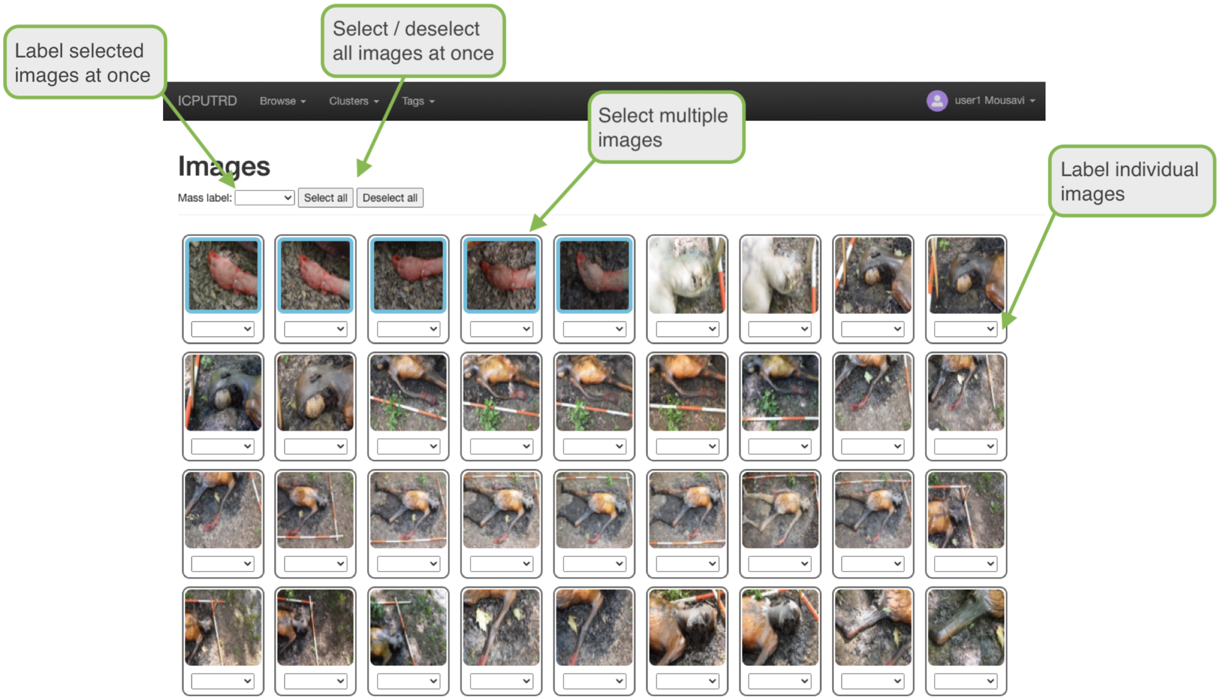


Figure 4.3: The image level labeling interface is shown. It allows users to label individual, multiple, or all images at once.

Classification

We used the labeled data resulting from the clustering-based approach to train a classifier model. The output of our classifier for each input is a feature embedding which is resulted from the convolutional layers only, a label and a confidence value for the given label. The initial training data for the classifier is obtained from the labels given by the expert to the initial clusters. During subsequent iterations, the classifier is used to make predictions on the batches of unlabeled data, and the predictions are ordered by the confidence level the classifier assigns to each prediction. High-confidence predictions are used to expand the training data using the predicted labels and low confidence predictions are assigned for clustering and manual labeling. The threshold for considering high or low confidence values is determined by manually exploring the predictions. The manual exploration is also used to prevent using labels that are wrong but are predicted with high confidence.

Adding images with high confidence predicted labels to the classifier's training data improve its accuracy on those types of images. Manual labeling of the images with low confidence predictions is still necessary to accurately predict the class for the instances where the certainty of the classifier was low. In other words, the low confidence values indicate the types of data the classifier has not yet fully learned. To expose the classifier to such data, we use the feature embedding outputted from the classifier for these images to cluster them. The clusters are then displayed to a domain expert for labeling and verification. Note that these embeddings are better representations of the images than those from the initial ImageNet-trained model as they have been partially trained on the domain data.

4.2.2 Evaluation and Experimental Results

We tested PLUD on a subset of the human decomposition dataset. We selected and labeled 5555 images capturing all images taken from 3 subjects as our ground truth data for testing PLUD. We used this test set to evaluate PLUD's classifier.

Furthermore, we evaluated the efficiency gained from our developed labeling interface in addition to the effects of data sampling and the size of training data on the labeling performance.

Classifier Performance

We analyzed the effect of adding data in each iteration on the performance of the classifier to ensure that in each iteration we have a more accurate model and thus less effort on the domain expert for manual labeling. We used three different models to find the model with highest accuracy. We checked the performance of VGG16, ResNet50 and Inception-based classifiers trained on various amounts of data, $\{1000, 2000, \cdots, 13000\}$. In each iteration, a new set of 1000 unlabeled images were fed to the classifier. High confidence predictions were used as new labels and images with low confidence predictions were labeled through the clustering interface. The new 1000 labeled images were added to the training data.

When the classifier was trained, its performance on a test set of 5555 manually labeled images was examined. The vall_acc, test_acc and F-score are shown in Figure 4.4. Increasing the size of the training data in all three classifiers resulted in an overall increase in their performance on our test data. It is important to note that although this result is expected, in the context of PLUD that means less manual effort on labeling for the domain expert since less misclustered images need to be manually selected each time.

Based on the results shown in Figure 4.4, we obtained the best accuracy when using an Inception-based classifier which was then used to calculate precision and recall (Table 4.1).

Labeling Interface Performance

To examine how our web interface affects labeling speed, we used 4 batches of 100, 200, 300, and 400 images and labeled each batch using a baseline interface and PLUD's. For the baseline method, we used the same interface as PLUD, but for each batch, users were presented with a pool containing all images of the batch, whereas, with PLUD, users were presented with clusters of images. Figures 4.5 and 4.6 show examples of the PLUD and baseline labeling processes. Figure 4.7 illustrates the amount of time spent on labeling the 4 image batches using the baseline and PLUD interfaces. The plot shows that our interface reduces the labeling time. It also shows that a linear increase in the number of images non-linearly increases the time of labeling. The result also indicates that labeling smaller

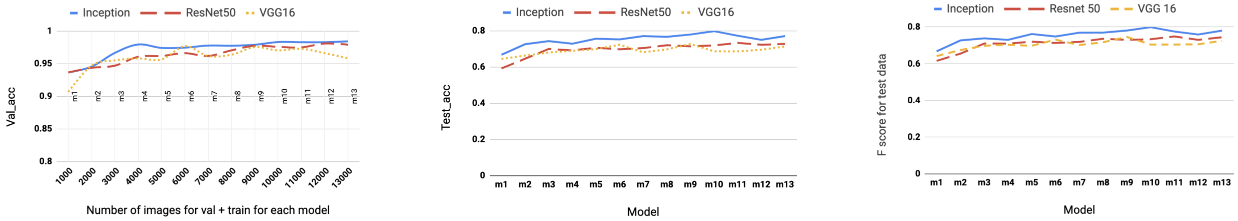


Figure 4.4: The performance of models trained on various number of images is shown. Test_acc and F score are calculated using the manually labeled test data. $\{m_1, m_2, \cdots, m_{13}\}$ are the models that are refined with more data through 13 iterations. Although it is expected to have better performance as the amount of training data increases, it is important to note that, the more accuracy in each iteration results in less effort for the user in the labeling process as shown in Figure 4.7.

Table 4.1: Shows precision and recall for PLUD, when using an Inception-based classifier, on the test data.

Model		**Precision of Classes**										**AP**
		Arm	**Hand**	**Foot**	**Legs**	**Full Body**	**Head**	**Backside**	**Torso**	**Stake**	**Plastic**	
Inception	Top 1	45.73	85.60	93.72	60.52	92.69	94.33	68.25	87.22	96.30	73.61	**79.80**
	Top 3	80.23	96.86	97.75	86.96	97.53	98.29	89.57	97.20	98.87	88.88	**93.21**

Model		**Recall of Classes**										**AR**
		Arm	**Hand**	**Foot**	**Legs**	**Full Body**	**Head**	**Backside**	**Torso**	**Stake**	**Plastic**	
Inception	Top 1	53.88	67.36	62.34	97.60	77.21	94.91	55.84	91.97	98.86	1	**80.00**
	Top 3	92.69	87.63	90.66	99.57	96.34	99.75	81.81	96.27	1.	1.	**94.47**

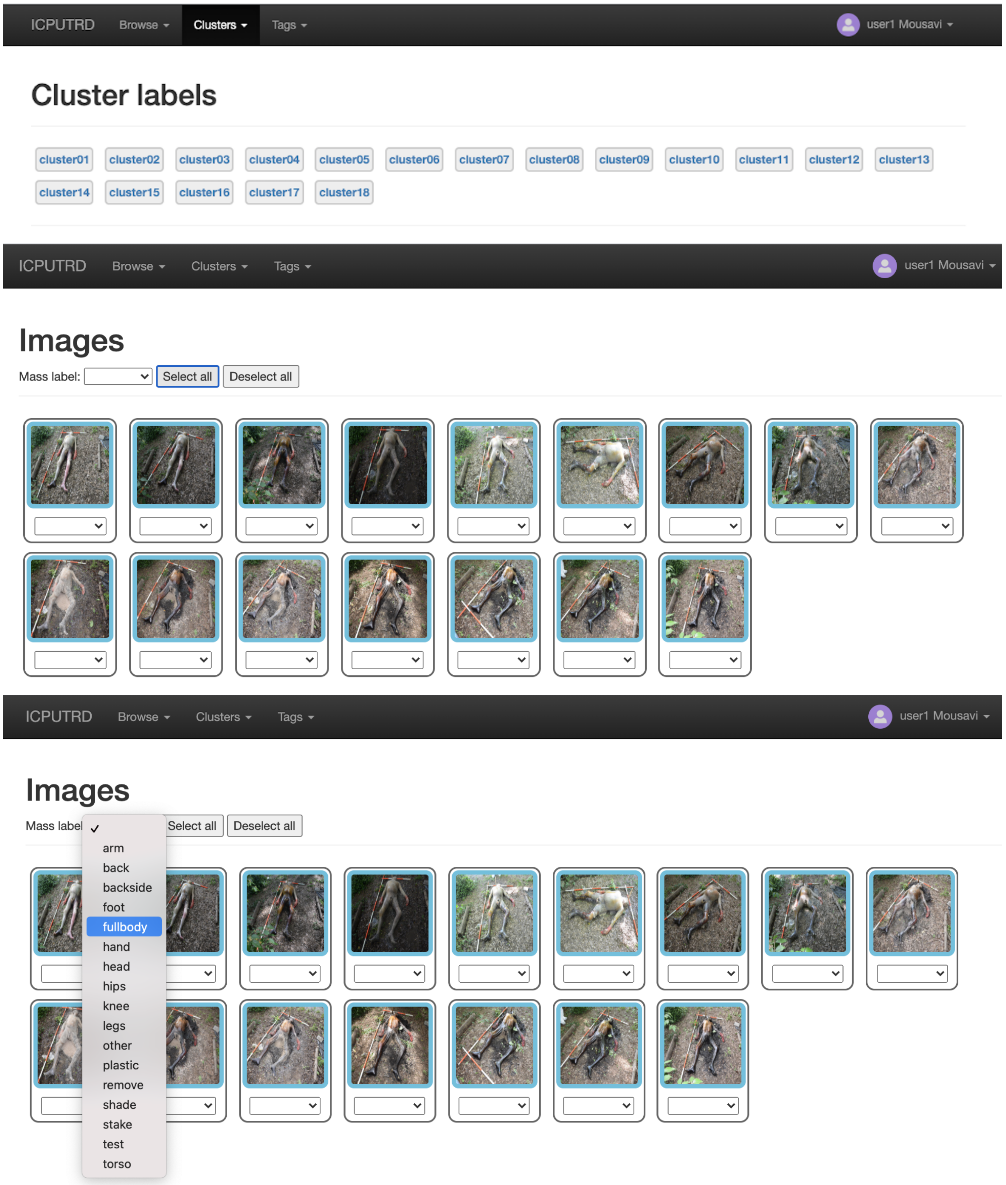


Figure 4.5: PLUD's labeling interface. Users are presented with clusters of similar images. Users can label the misclustered images first and then label the remaining images all at once, which can potentially result in less labeling effort. The better the clusters are, the less labeling effort is required.

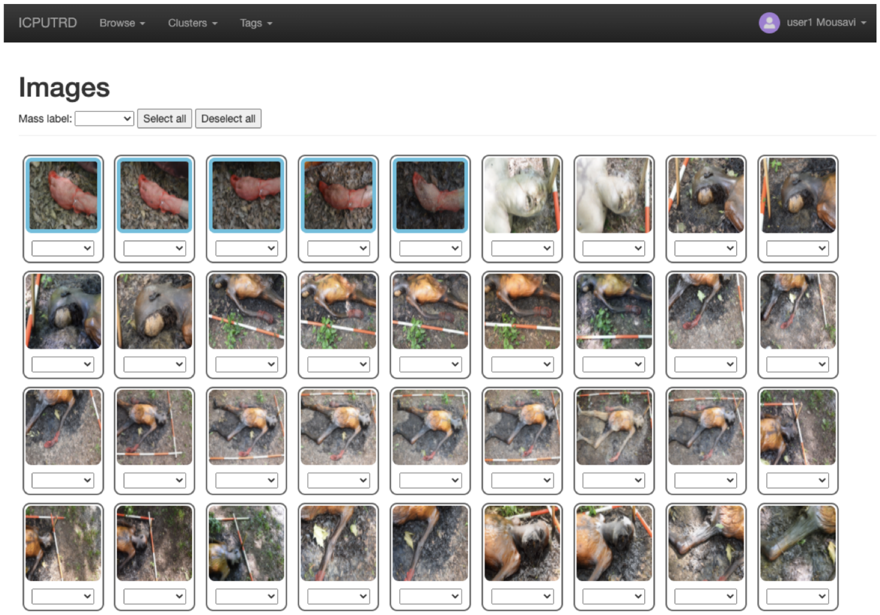


Figure 4.6: The baseline interface is shown. The user is presented with a pool of images to be labeled. The interface allows the user to label images one by one or multiple image at the same time.

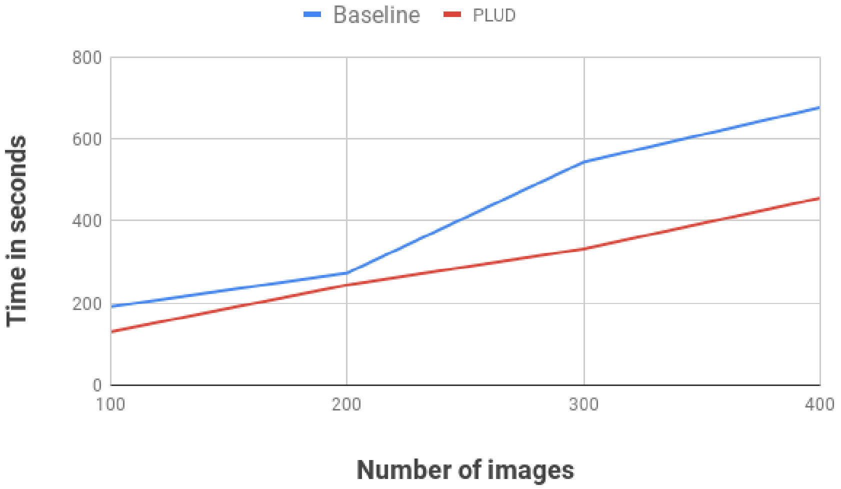


Figure 4.7: Comparing the amount of time spend on labeling 4 image batches using the baseline and PLUD interfaces.

batches of data is faster. We believe that this is due to the visual confusion caused by scrolling through a large number of images.

Data Sampling Strategy

It is important to make sure that the training data is diverse and represents the structure of the dataset. We designed an experiment to test this hypothesis by having a set of 5000 randomly selected images from the entire dataset and 5000 images from multiple randomly selected subjects while making sure it includes all stages of decomposition. We trained our classifier on the two sets and tested it on our test set. The results indicated that the classifier trained on the structure-aware multi-subject data performed with 76.71% accuracy on the test set where as the accuracy was 74.51% for the classifier trained on randomly selected data. This confirms the importance of meaningful sampling for training the model.

4.3 SLRNet: Similarity-based Label Reuse for Semantic Segmentation

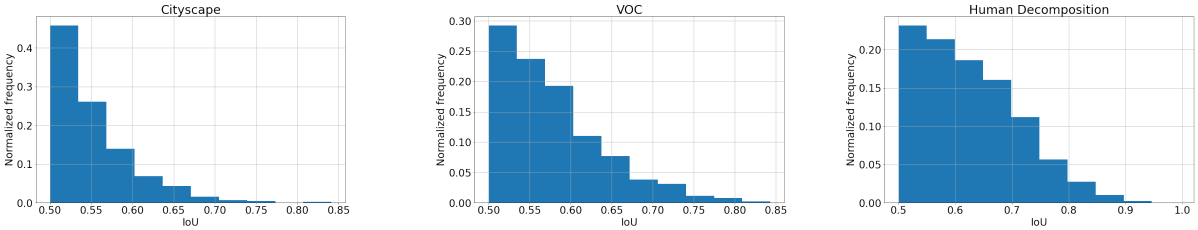
Image segmentation is an important computer vision task that assigns labels to images at pixel-level granularity. Pixel-level labels provide information representing the objects depicted by the images and their exact locations in the image. Image segmentation is needed for any application that requires knowledge of the coordinate of objects in images.

Pixel-level annotations are necessary for accurate supervised semantic segmentation but may be too costly for many application domains. Manual pixel-level labeling needed for semantic segmentation can take 15 to 60 times longer than that of region-level and image-level labels [81]. In addition, specialized expertise is often necessary and scarce for labeling images in domains such as medicine or forensic anthropology. To address such challenges, many semi-supervised and weakly-supervised methods have been developed to work with relatively few labeled and numerous unlabeled or weakly labeled images.

Weakly-supervised segmentation methods [45, 76] rely on weakly annotated data to produce annotations for the unlabeled portion of the data. The newly produced annotations along with the original existing ones are then used in a supervised manner for the task of interest. However, due to the need for weak labels in such methods, recently, semi-supervised segmentation methods have gained more attention. Many semi-supervised segmentation methods have been developed mainly based on generating more data using GAN-based methods, producing pseudo-labels for existing unlabeled images [74, 11, 124] or through consistency-based methods utilizing augmentations [73, 93, 129], perturbations [73], and multi-model collaborations [66]. While many of these methods have complex structures and their results highly depend on well-tailored perturbation techniques, in this work, we explore the possibility of exploiting intrinsic differences and similarities in the data itself.

Knowing the structure of evolving data and their gradual changes caused by evolution, such datasets are more likely to include images that can potentially have similar annotations (pixel-level labels). To confirm this hypothesis, we experimented on the VOC [43], Cityscape [28] and the human decomposition datasets. We calculated the intersection over the union (IoU) between the labels for pairs of images with the same classes for each dataset. Histograms in Figure 4.8 show the result for this experiment which confirms the hypothesis that there exist more label similarity in the evolving image dataset. In this dissertation, we leverage this characteristic to reuse the human-provided labels as pseudo-labels for unlabeled images that potentially have similar labels to them. We then use a combination of the labeled and pseudo-unlabeled images to train a semantic segmentation model. As is expected, the location and orientation of the objects captured in the unlabeled images may not be perfectly aligned with that of the labeled ones. Therefore, in our semantic segmentation network, we introduce a custom loss function to control the impact of the pseudo-labels on the learning process.

We present a semi-supervised method to tackle the semantic segmentation problem by exploiting latent relationships among images. Our algorithm discovers such latent relationships in order to obtain pseudo-labels for unlabeled images by first pairing the labeled images with similar unlabeled images and then reusing available pixel-level annotations for similar but unlabeled images in each pair. A customized loss function is then used to penalize the prediction errors depending on the level of annotation similarity between images in each pair. In other words, the key idea is to exploit the similarity present in



(a) (b) (c)

Figure 4.8: Histogram of IoU between pairs of labels for images with the same classes for Cityscape, Pascal VOC, and the human decomposition datasets are shown. The plots indicate that in a dataset with evolving content such as the human decomposition dataset, there exist more label similarity than others.

the image collection by re-using annotations for unlabeled images weighted by the extent of the network's understanding of their similarity while also jointly learning from supervised samples. We call our method SLRNet for Similarity-based Label Reuse Network[3] [4] to support such tasks.

While many large image datasets may contain groups of similar images satisfying the key assumption of the proposed approach, we focus on evaluating our method on the human decomposition dataset which is very important for forensic anthropology researchers.

Segmenting body parts in the images of human decomposition is very important for forensic research for two reasons. First, it enables separating body parts from the background area and therefore, facilitates the identification of other forensic features. Second, the decomposition type varies among body parts [122, 51], hence the same forensic feature needs to be placed in context to pave the way for downstream models of time-of-death estimation that include forensic features and body parts as key predictors [53, 18, 14].

Semantic segmentation of decaying body parts is complicated due to various reasons. First, the color and texture of decaying body parts and many forensic features such as mummified skin being extremely similar to the background (see Figure 4.9). Second, in human decomposition imagery, features gradually evolve over time from the "fresh" stage to the completely decayed stage of "skeletonization". Third, different body parts are difficult to distinguish due to environmental settings such as muddy ground, especially in late stages of decay. Finally, the often multi-day delay between photos, the lack of control in the camera view, changing weather conditions, disturbances caused by scavenging, and the lack of explicit links over time of classes limits the applicability of powerful video object segmentation (VOS) methods. However, the gradual decay of these subjects brings about a *similarity* attribute in this dataset that can be utilized toward developing an efficient but simple semi-supervised method. We present SLRNet which is a simple method that reuses labeled data as pseudo-labels for unlabeled images. Specifically, we first use an unsupervised algorithm to identify immediately or transitively similar images to each labeled image. This

[3] Link to the repository: https://github.com/saramsv/SLRNet

[4] Mousavi, Sara, et al. "Similarity-based Label Reuse for Semi-Supervised Semantic Segmentation of Human Decomposition Images." Submitted to proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022.

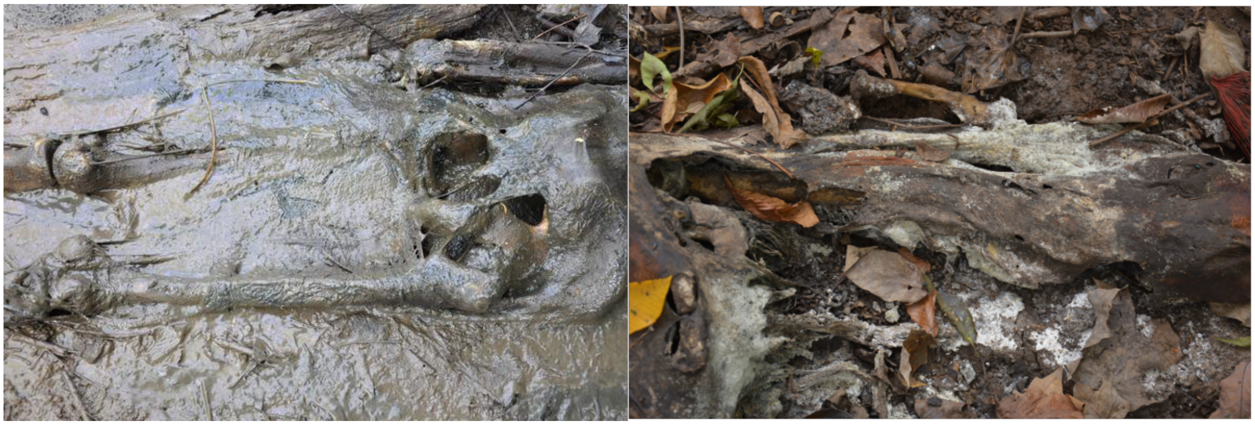


Figure 4.9: Examples of the background having a similar color and texture to the foreground in the human decomposition dataset.

algorithm exploits the fact that the images depicting the same evolving subjects are more similar if they are closer to each other on the decay spectrum, and the same subject may have a different appearance in early and late evolution stages, however have very similar annotations.

We find similar images to a given labeled image among its neighbors recursively until no image with sufficiently high similarity is found. Pairs are then created between the annotated image and the similar images, with the annotation of the labeled images being applied to the unlabeled images as pseudo-labels. However, when training the semantic segmentation network, we do not treat the losses from pseudo labels equally, but adjust them based on the level of similarity between the network's predictions for the images in the pairs.

We evaluate our method on the human decomposition images and compare our method with two state-of-the-art semi-supervised semantic segmentation methods: CCT [104] and PseudoSeg [154]. Results indicate that SLRNet, while having a much simpler conceptual structure and correspondingly shorter runtime, outperforms both CCT and PseudoSeg on evolving images of human decomposition. Figure 4.10 illustrates our method being applied on an image of a decomposing foot, along with its ground-truth label and prediction using SLRNet. Additionally, we also test SLRNet against a dataset of plant growth (Aberystwyth Leaf dataset) which exhibits similar characteristics to the human decomposition data.

In the following, we provide details about the structure of SLRNet and its performance in segmenting images from the human decomposition and the Aberystwyth Leaf datasets.

4.3.1 Architecture Design

Figure 4.11 shows an overview of our method, which consists of two main steps: image-pairing and network training. In the pairing step, we use an unsupervised method, similar to [98], to identify unlabeled images that are similar to the labeled images, and therefore, can be paired with them. In the training phase, first, we simply reuse the annotation of the labeled images by assigning them as pseudo-labels to their similar but unlabeled match. We then use the image pairs along with the original labeled images as a separate set to train our semi-supervised semantic segmentation network in an end-to-end fashion. Our network is fed with both labeled images and the pairs at each iteration to jointly learn from both

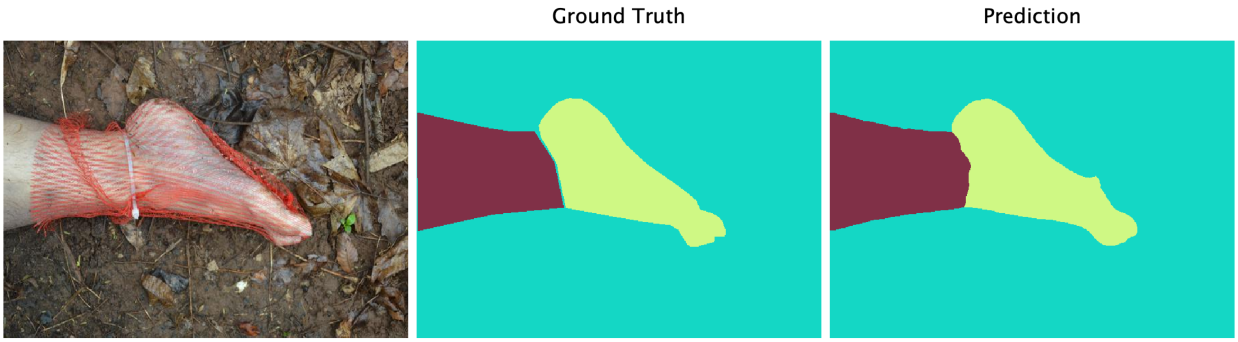


Figure 4.10: An example of semantic segmentation for images of human decomposition using SLRNet.

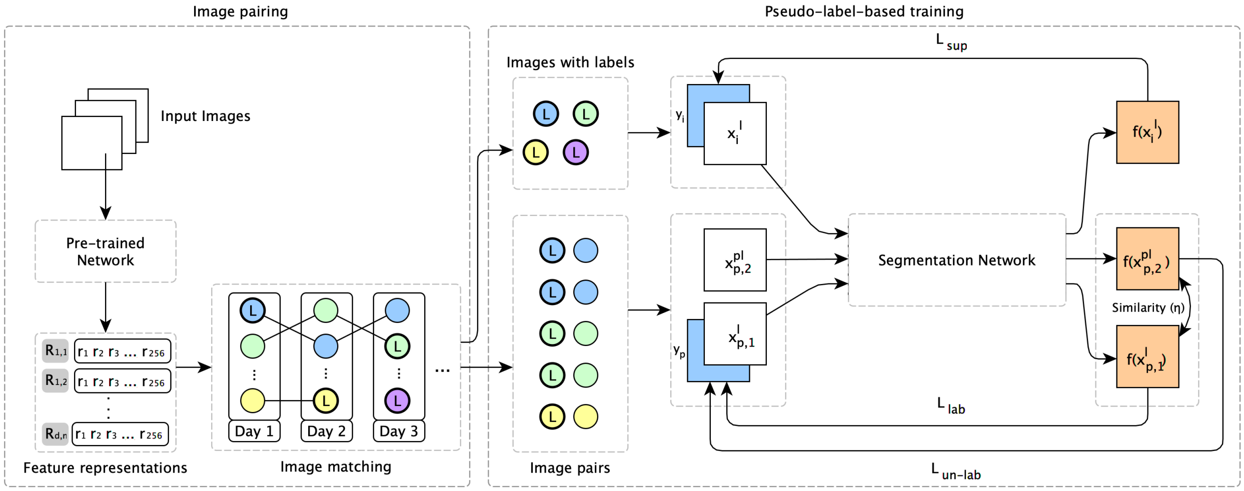


Figure 4.11: An overview of our method is shown (best seen in color). Images, first, go through a classification network which is pre-trained on ImageNet [37] and fine-tuned on the human decomposition data to be mapped to feature vectors. Labeled images (shown with circles labeled with 'L') and unlabeled images (circles with no labels) along with their feature vectors are then fed to our unsupervised image matching component to obtain similar images that can be paired with labeled ones and reuse the labels as pseudo-labels for the unlabeled images in the pairs. These pairs (one labeled image and one unlabeled image) along with the labeled images as a separate set, then go through our segmentation network that uses a customized loss function to control the effect of the pseudo-labels in the training process based on the similarity between the network's predictions (η) for the pair. The network minimizes the sum of losses calculated based on the labeled images (l_{sup}) and the pairs ($l_{lab} + \lambda \times \eta \times L_{un-lab}$). λ and η are used to control the impact of pseudo-labeled images on the network's learning.

images with original labels and pseudo labels. Throughout the training, however, we use a custom loss function to control the effect of the unlabeled images and their pseudo-labels on the learning process with respect to the similarity of the two predictions for the images in the pairs as well as the current learning iteration. In the following sections, we provide details on the unsupervised image pairing process, the network structure and its training process, and evaluation and experimental results for SLRNet.

Unsupervised Image Pairing

The main idea of pairing labeled images with unlabeled ones is to utilize the potential reuse of annotations. It is likely for a dataset to have two or more images depicting the same content with only small differences in their labels. This is even more the case for image datasets with evolving content such as images of human decomposition, aging faces, growing plants, or decaying produce.

In the human decomposition dataset, while photos from the same subjects at early evolution stages can drastically differ from those at late stages in terms of visual features, they still tend to have similar annotations. For example, while a hand in the "fresh" stage might look different from a hand in the late stages of decay, their annotations still resemble a hand from the same body and have similarities. Moreover, images belonging to neighboring days are more likely to be similar as well. We leverage this characteristic and use neighbor-based comparison to identify the most suitable pairs for a given set of labeled images.

First, we use the feature maps of a pre-trained classification network such as ResNet to map all images to their corresponding feature representations. To do so, we feed the images into the classification network, excluding its last fully-connected and softmax layers, and use the output vectors as the feature representations. In this work, we used ResNet50 [58] pre-trained on ImageNet [37] and fine-tuned with human decomposition data. This fine-tuning uses only image-level labels and increases the probability that similar images would contain the same body part. Other CNNs such as Inception [128], and VGG [123] may also be used. In the case of using ResNet, the vector length is 2048. Inspired by DeepCluster [20], we also reduce the length of these representations to 256 using Principle Component Analysis [138] to improve the overall runtime of our method. We denote the resulting feature representation for $I_{d,n}$ which is the n^{th} image in day d, as $R_{d,n}$.

To identify potential images to pair with the labeled image $I^l_{d,n}$ (marked with an l superscript to denote that it is labeled), we compare its feature representation (i.e. $R^l_{d,n}$) with all other feature representations of its neighboring days and pick the most similar image as a match. From there, we recursively compare the matched images with their own respective neighbors to find other suitable pairs for the original $I^l_{d,n}$. The higher the neighboring days, the larger the pool of images to find the match from, and as a result, the more computation we will need. In this work, we use 4 as the number of days considered as neighboring days for a given image. Other values can be used as well.

While this process can track similar images throughout different stages of decay, it also forces a match between each two neighboring days even if the difference is large. To circumvent this issue, we stop the matching process if the similarity of the two images being compared is lower than the overall average similarity calculated up to that point among the potential matches for $I^l_{d,n}$. Comparison of the image features is done through cosine similarity as:

$$Similarity(R_{d,n}, R_{d',m}) = \frac{R_{d,n}.R_{d',m}}{\|R_{d,n}\| \, . \, \|R_{d',m}\|} \tag{4.1}$$

The pairing process is detailed in Alg. 1 where, for each labeled image, initially the *"Compare"* function is called for it and itself as *"ref"*. In the *"Compare"* function, each reference image (*"ref"*) is compared to all of its neighbors and images that are found to be appropriate matches will be added to the list of images that can be paired with the labeled image (*"l"*). The *"Compare"* function is then recursively called for each image in the match set.

From the matches obtained for $I^l_{d,n}$, a few may coincidentally have labels. Those are removed from the group of images being paired with $I^l_{d,n}$ as they will be later paired with unlabeled matches of their own. A few example pairs are shown in Figure 4.12.

```
Result: Pairs of labeled and unlabeled images
Def Main():
    matches = {}
    for each labeled image l do
        Compare(l, l)
    end
    pairs = []
    for each labeled image l in matches do
        for each unlabeled match ul in matches[l] do
            pairs.append(l, ul) ;
        end
    end
Def Compare(l, ref):
    R = PCA(Feature(ref))
    for each neighboring day D for image ref do
        maxSim = 0
        match = NULL
        for each image $i'$ in D do
            R' = PCA(Feature(i'))
            s = Similarity(R, R')
            if $s > maxSim$ then
                maxSim = s
                match = i'
            end
        end
        if $maxSim \geq AvgSim(matches[l])$ then
            matches[l].add((match, maxSim))
            Compare(l, match)
        end
    end
```

Algorithm 1: Image matching algorithm

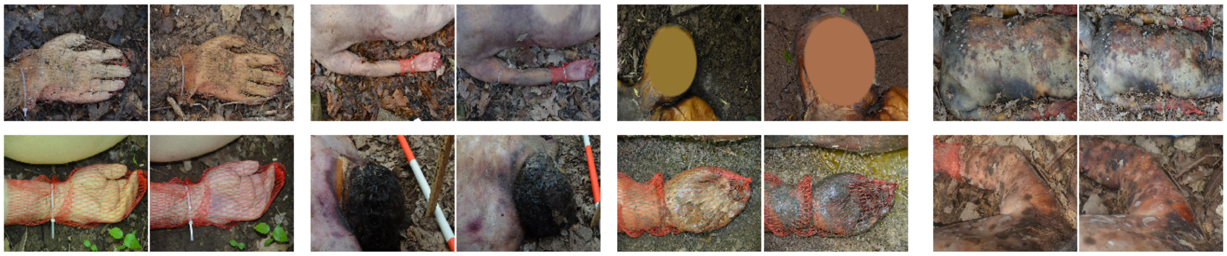


Figure 4.12: A few pair examples generated using our pairing algorithm for images from the human decomposition dataset. The third pair from the first row is anonymized.

Network Structure and Training

The set of labeled images is denoted by $\mathcal{I}^l = \left\{(x^l_1, y_1), (x^l_2, y_2), \cdots (x^l_L, y_L)\right\}$, where L is the total number of labeled images, y_i is the annotation for the i^{th} labeled image (i.e. x^l_i) and has dimensions of $W \times H \times C$ representing width, height and the number of classes, respectively.

We reuse the annotations of the labeled images by assigning them as pseudo-labels (marked with a *pl* superscript to denote that they have pseudo-labels) to the image with no labels in the pairs obtained from our image pairing algorithm described in the previous section. We denote the pairs with: $\mathcal{P} = \left\{\left((x^l_{1,1}, x^{pl}_{1,2}), y_1\right), \left((x^l_{2,1}, x^{pl}_{2,2}), y_2\right), \cdots \left((x^l_{P,1}, x^{pl}_{P,2}), y_P\right)\right\}$. where P is the total number of pairs and $x^l_{p,1}$ and $x^{pl}_{p,2}$ are the first labeled and second pseudo-labeled elements in the pth pair respectively. In this setting $\left\{x^l_{1,1}, x^l_{2,1}, \cdots, x^l_{P,1}\right\} \subset \mathcal{I}^l$ and these are not unique images, whereas $\left\{x^{pl}_{1,2}, x^{pl}_{2,2}, \cdots, x^{pl}_{P,2}\right\}$ are P unique unlabeled images. That is because a labeled image could be paired with more than one image.

It is important to note that there may not exist an unlabeled match for every single image in the labeled set. Therefore, we feed the labeled images in a separate set, parallel to the pairs, even though some of them might be already included in the pair branch, to ensure the inclusion of all labeled images in the training process.

Additionally, as is expected, the location and orientation of the body parts captured in an unlabeled image may not be perfectly aligned with that of the labeled image in the pair. Therefore, we introduce a custom loss function to control the impact of the pseudo-labels on the learning process with respect to the level of similarity between the network's predictions for the two images in the pair.

Our objective is to exploit the additional pseudo-labeled images to improve the performance of the semantic segmentation network. In this method, we use a semantic segmentation network with a custom loss function to facilitate learning from both labeled and pseudo-labeled images. At each iteration, the network is fed with a labeled image and a pair of images. We calculate a supervised loss and a loss for pairs, following Eq. 4.2 and Eq. 4.3 respectively. In the supervised loss calculated for the labeled images (Eq. 4.2), CE is cross entropy calculated using the ground truth y_i and the predicted labels for input x^l_i which is shown by $y^{'}_i = f(x^l_i)$.

$$\mathcal{L}_{sup} = \frac{1}{|\mathcal{I}^l|} \sum_{x_i^l, y_i \in \mathcal{I}^l} CE(y_i, y_i^{'}) \tag{4.2}$$

Next, we calculate a loss for the pairs. As mentioned above, each pair has one labeled and one unlabeled image. We calculate pair loss following Eq. 4.3.

$$\mathcal{L}_{pair} = \mathcal{L}_{lab} + \mathcal{L}_{un-lab} = \frac{1}{|\mathcal{P}|} \sum_{x_{1,i}^l, y_i \in \mathcal{P}} CE\big(y_i, f(x_{1,i}^l)\big) + \frac{1}{|\mathcal{P}|} \sum_{x_{2,i}^{pl}, y_i \in \mathcal{P}} \lambda * \eta * CE\big(y_i, f(x_{2,i}^{pl})\big) \tag{4.3}$$

where $i \in \{1, 2, ..., P\}$. The first part of the loss is calculated based on the labeled image in the pair, y_i and $f(x_{1,i}^l)$, and the second part is calculated based on the unlabeled image in the pair, using y_i and $f(x_{2,i}^{pl})$. Since y_i are not actual labels for $x_{2,i}^{pl}$, the loss calculated based on them is weighted by a similarity-based weight, η, and an iteration-based weight, λ. We use the prediction of the network for the pair, to determine the level of contribution for the loss calculated based on the pseudo labels in the backpropagation and calculate η accordingly. The idea is to use the network's understanding to measure how similar and aligned the annotations for the images in a pair are. We compare the prediction of the network for the images in the pair to each other. The higher the similarity, the larger the weight will be. The value of η is at most 1 and calculated as:

$$\eta = \frac{f(x_{1,i}^l) \cap f(x_{2,i}^{pl})}{(W \times H)_{x_{1,i}^l}} \tag{4.4}$$

where W and H are the width and the height for $x_{1,i}^l$ respectively. Furthermore, $f(x_{1,i}^l)$ and $f(x_{2,i}^{pl})$ are the predictions of the network for $x_{1,i}^l$ and $x_{2,i}^{pl}$ respectively.

Additionally, since the network's prediction is not robust at early epochs, we use λ calculated based on the iteration numbers to minimize the influence of the initial noisy predictions and incorrect use of the pseudo labels in the training process. The value of λ linearly increases with training iteration and is at most 1.

$$\lambda = \frac{(epoch * ipe + iter)}{max\_iters} \tag{4.5}$$

where *epoch* is the current epoch, *ipe* is the number of iterations per each epoch, *iter* is the current iteration, and max_iter is $the\ total\_number\_of\_epochs \times ipe$.

The network is trained to minimize the overall loss that is calculated as:

$$\mathcal{L} = \mathcal{L}_{sup} + \mathcal{L}_{pair} \tag{4.6}$$

4.3.2 Evaluation and Experimental Results

In this section, we provide details on the implementation of our proposed method, train and test data as well as the evaluation metrics, comparison to state-of-art methods, and an ablation study on different parameters of our method.

Implementation Details

We implemented our method using HRNet[127], Xception [24] and ResNet [58] as the backbones. Other networks can also be used. The HRNet version used in SLRNet is HRNetV2. We implemented SLRNet using the PyTorch framework [105]. We trained our method on a single $TeslaV100{-}SXM2$ GPU with $32GB$ memory. To train the segmentation network, we used Stochastic Gradient Descent (SGD) [67] with momentum of 0.9 and weight decay of 10^{-4}. We started with a learning rate of 0.02. The learning rate is gradually decreased using polynomial decay with power 0.9 [22]. We used the number of train samples divided by batch size as iterations per epoch.

Evaluation Metrics and Datasets

Similar to other semantic segmentation works [60, 104, 125], we use the mean intersection-over-union (mean IoU) and pixel accuracy as evaluation metrics.

We evaluate SLRNet on the human decomposition images. This dataset includes 1864 annotated images for “hand”, “arm”, “foot”, “leg”, “torso”, and “head” classes. We use $60\%, 20\%, 20\%$ ratio to create training, validation and test sets. As a result, we have 1118, 373, and 373 images in our training, validation and test sets respectively. Additionally, we use 5906 unlabeled images resulted from identifying matches for the labeled images in the training set with other unlabeled images in the dataset using our pairing algorithm.

In addition, we evaluate the generality of our proposed method by conducting similar experiments to the human decomposition on a dataset from a different domain capturing growing plants (Arabidopsis leaves called Aberystwyth Leaf Evaluation Dataset [9]). Image data depicting different stages of growing plants manifest some similarities to the human decomposition photos as both are depicting gradual changes over time that, over the full course of observation, lead to dramatic changes in appearance, but provide local similarities between neighboring timesteps that our method can leverage.

The Aberystwyth Leaf dataset records the growth of Arabidopsis Thaliana plants potted in four trays. This dataset includes 134040 Arabidopsis Thaliana plants, from which 916 are manually annotated. We use the same ratio as the human decomposition data, 60%, 20%, 20%, to create training, validation, and test sets respectively. Using our pairing algorithm, we use an additional 31440 individual Arabidopsis plants paired with the labeled ones in the training process. A few paired examples are shown in Figure 4.13. In our experiments with this dataset, we set the number of classes to two, for “background” and “leaf”.

Comparisons to Previous Work

To the best of our knowledge, we are the first to explore semantic segmentation for images with evolving content such as human decomposition data and therefore there are no similar benchmarks or state-of-the-art methods on this topic. Therefore, to evaluate the effectiveness of our method, we quantitatively compare it with previous general state-of-the-art semi-supervised semantic segmentation methods on PASCAL VOC dataset [44], called CCT [104] and PseudoSeg [154].

CCT is a consistency-based method that enforces consistency to the network’s predictions for various perturbed version of an input. It uses a two branch training structure one for labeled and one for unlabeled data. The two branches share the same encoder and one decoder. On the unsupervised branch it uses $K = 7$ auxiliary decoders and various perturbations (minimum of 2 and maximum of 6) and enforces consistency between their

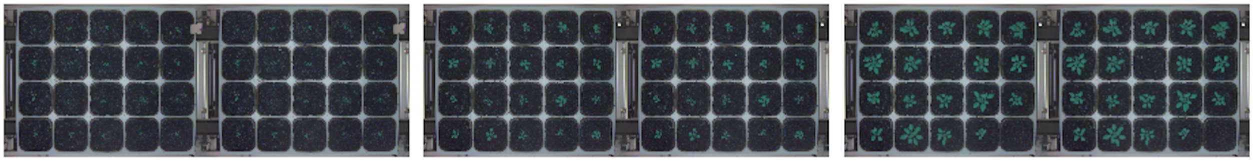


Figure 4.13: A few pair examples generated using our pairing algorithm for images from the Aberystwyth Leaf dataset.

outputs and the output from the main shared decoder on the same input through a loss function.

PseudoSeg uses a mix of pseudo-labeling and consistency-based training to leverage unlabeled images in the network's learning [154]. In PseudoSeg, the pseudo-labels are generated by fusing the network prediction for a weakly augmented input image and the self-attention GradCAM generated for that input. In the training process, the authors use a similar idea to consistency-based methods and force their network's prediction for a strongly augmented version of the same input to be consistent with the pseudo-label that resulted from the fusion process.

To compare our method to CCT and PseudoSeg, we applied their semi-supervised setting to our data and compared the results with those obtained using our method. CCT and PseudoSeg use both labeled and unlabeled images to learn from them jointly. We use the unlabeled images obtained from our pairing algorithm and the labeled images as their unlabeled and labeled training inputs, respectively. We use the same validation and test sets for our method, CCT, and PseudoSeg. The results of comparing our method to CCT and PseudoSeg are shown Table 4.2.

The results indicate that our method outperforms both CCT and PseudoSeg using the mean-IoU and mean-Acc metrics with a few minor exceptions for "BG", "Arm", and "Head" classes in PseudoSeg. The results also confirm that, in general, the HRNet backbone captures spatial information better and performs well even for classes that occupy fewer pixels than the others and outperforms the other backbones for all methods. In the human decomposition data used in this work, this difference is pronounced due to the need to maintain multiple feature resolutions since we have images of the same class with varying views. Furthermore, the run-time reported in Table 4.2 shows that SLRNet has a shorter run time than CCT and PseudoSeg while having a conceptually simpler structure. For the Aberystwyth Leaf dataset, we conducted two sets of experiments. First, to see the impact of the additional unlabeled data on the performance of our method in segmenting the Aberystwyth Leaf images, we compared the supervised version, when the model is only trained on the labeled data, vs. a semi-supervised version, when the model is trained on both labeled images and the pairs generated using our pairing algorithm. Second, we compared the performance of

Table 4.2: Mean-IoU, mean-pixel accuracy, per class IoU, and run-time for CCT, PseudoSeg and SLRNet on the test data. The results indicate that our method consistently outperforms other methods on most classes with a large margin. Results are in percentages.

	Backbone	**mIoU**	**mAcc**	**Per Class IoU**							**Run time**
				BG	**Foot**	**Hand**	**Arm**	**Leg**	**Torso**	**Head**	
CCT	ResNet	62.77	86.67	88.73	52.16	53.29	48.56	63.39	60.67	72.57	23967m26s
PseudoSeg	Xception	62.22	86.62	88.45	48.47	49.11	43.88	61.51	63.93	80.16	1334m5s
	ResNet	62.49	86.99	88.56	45.84	47.16	50.46	62.74	64.04	78.59	1500m17s
Our Method	Xception	66.76	88.27	88.63	78.63	78.63	78.63	78.63	78.63	78.63	1300m19s
	ResNet	65.3	87.12	87.44	57.28	58.07	49.68	64.32	64.75	75.59	764m15s
	HRNet	**72.42**	**90.04**	**90.01**	**62.42**	**67.9**	**61.06**	**69.5**	**72.06**	**83.98**	708m24s

our method to that of two state-of-the-art methods, CCT [104] and PseudoSeg [154] using the ResNet backbone for a fair comparison to CCT.

The result of the two experiments are shown in Tables 4.3 and 4.4. The results in Table 4.3 indicate that our method (SLRNet) indeed facilitates controlled use of the additional unlabeled data and the semi-supervised setting that includes this data performs noticeably better as compared to the supervised version. Furthermore, table 4.4 shows the result of comparing SLRNet with CCT and PseudoSeg. The results indicate that our method consistently outperforms both CCT and PseudoSeg methods in all metrics. Notably, PseudoSeg and SLRNet perform much better on the plant dataset. This is not particularly surprising since only two classes are predicted (leaf and background) for plants in contrast to a much harder problem of 7 classes for human decomposition.

Additionally, CCT performed considerably worse than the other two methods in terms of its mean IoU. We suspect that CCT's complex structure and its various perturbations hinder its prediction of small leaves.

Ablation Study

We present an ablation study of our proposed method on the human decomposition data in Table 4.5. We examine the impact of λ, η, and the additional unlabeled data produced using our pairing algorithm on the overall performance of our method. In addition, we analyze the impact of the quality of the pairs by replacing them with random ones (meaning each labeled image is paired with a random unlabeled image) and see how they affect the performance.

SLRNet, as a whole, uses both labeled and paired images, weights the loss calculated based on the pseudo-labels using η, and reduces the initial noisy and unstable behavior of the network using λ. We conduct the ablation study by assessing the performance of our method in the absence of these components. The scenarios shown in Table 4.5 are:

1. the presence of labeled images, pairs, λ, and η

2. the presence of labeled images, pairs, and η and the absence of λ

3. the presence of labeled images, pairs, and λ and the absence of η

Table 4.3: Supervised vs. semi-supervised comparisons of SLRNet with different backbones for segmenting images of Aberystwyth Leaf dataset. Results are shown in percentages.

	SLRNet			
	Supervised		**Semi-supervised**	
	mIoU	**mAcc**	**mIoU**	**mAcc**
HRNet	92.43	97.28	**94.27**	**98.03**
ResNet	88.17	95.64	**94.2**	**98.01**
Xception	63.53	83.94	**65.13**	**86.99**

Table 4.4: Mean-IoU and mean-pixel accuracy for CCT, PseudoSeg and SLRNet on the Aberystwyth Leaf test data using the ResNet backbone.

	Mean-IoU (%)	**Mean-Acc (%)**	**Per class IoU (%)**	
			BG	**Leaf**
CCT	51.73	82.59	81.71	21.75
PseudoSeg	90.64	96.7	95.92	85.35
SLRNet	**94.2**	**98.01**	**97.52**	**90.88**

Table 4.5: Ablation study to examine the effect of λ, η, the additional unlabeled data produced using our pairing algorithm, and the quality of the pairs on the overall performance of our method with different backbones. Results are in percentages.

					HRNet		**ResNet**		**Xception**	
	Labeled Images	Pairs	λ	η	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
Supervised	✓	✗	✗	✗	68.42	88.86	47.38	80.11	61.15	86.39
Semi-supervised	✓	✓	✗	✗	61.37	85.94	54.53	82.04	63.46	86.44
	✓	✓	✗	✓	63.97	86.24	60.71	85.05	63.65	**88.71**
	✓	✓	✓	✗	67.18	88.37	59.89	84.53	62.24	86.41
	✓	✓	✓	✓	**72.42**	**90.04**	**65.3**	**87.12**	**66.76**	88.27
Semi-supervised	✓	Random	✓	✓	69.43	89.38	58.53	85.81	62.85	86.84

4. the presence of labeled images, pairs and the absence of both λ and η

5. the presence of labeled images and the absence of pairs, λ, and η

6. the presence of labeled images, random pairs, λ, and η

When η is absent, it means there is no similarity-based weighting for the loss calculated based on pseudo-labels. That means the pseudo-labels are used as actual ground-truths and the network is basically trained in a supervised fashion on both labeled images and those in pairs. The scenario with labeled images and no pairs, no λ, and no η is equivalent to a fully supervised version of our method when it is only trained on the 1118 labeled images.

The results of the ablation study indicate that our method performs its best, in terms of mIoU, when we include the additional unlabeled data produced using our pairing algorithm as well as using both λ and η to avoid initial noisy prediction of the network and control the contribution of the pseudo-labels in the training process. In addition, we observe that the supervised version of HRNetV2 performs better than the semi-supervised version of other backbones, and the additional data from pairs without being carefully controlled by η and λ would hurt its performance. That is because HRNet maintains multiple high-resolution features in parallel throughout its training and exchanges information between them via a multi-scale fusion. HRNetV2 specifically outputs four-resolution representations containing rich and precise spatial information; therefore, it outperforms other backbones in supervised training, but it can be potentially more sensitive to noise if it is not controlled by η and λ. Furthermore, the results indicate that the quality of the pairs is also an essential factor on the performance of the network compared to using random pairs.

Furthermore, we qualitatively evaluate the performance of our method by providing the resulted segmentation for a few examples from the human decomposition and Aberystwyth Leaf datasets using CCT, PseudoSeg, and SLRNet. These examples are shown in Figures 4.14 and 4.15. The results indicate that, overall, our method captures classes and their annotations more accurately.

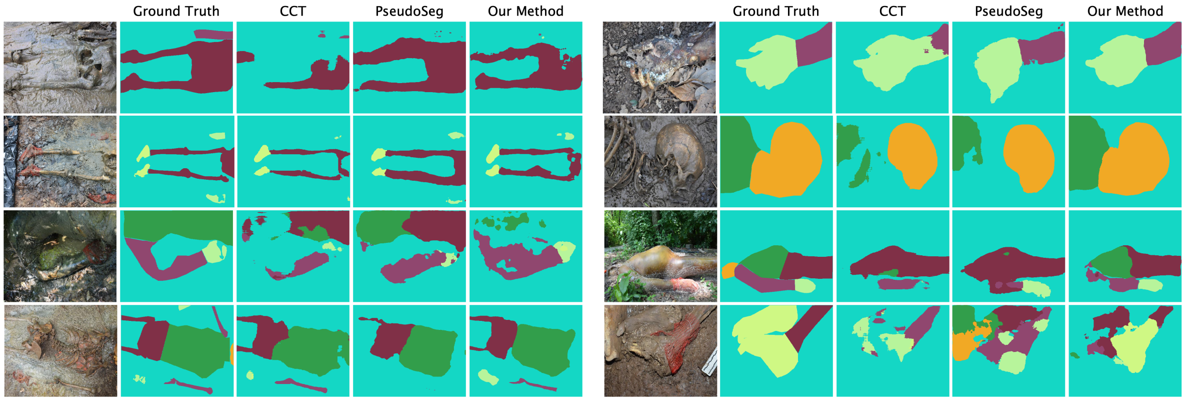


Figure 4.14: A few examples of our semantic segmentation method and those from CCT and PseudoSeg on the human decomposition dataset when all methods are trained on 1118 labeled images and 5906 unlabeled images.

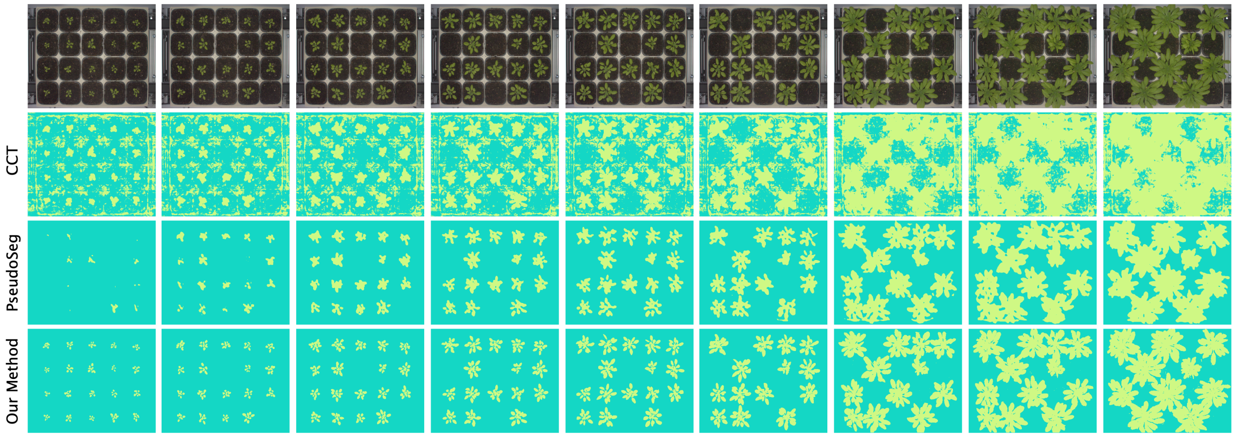


Figure 4.15: A few examples of performing semantic segmentation on the Aberystwyth Leaf dataset when using SLRNet, CCT and PseudoSeg. All methods are trained on 916 labeled images and 31440 unlabeled images from the Aberystwyth Leaf dataset.

Chapter 5

Data Dissemination

An organized, contextualized, and enriched dataset is only valuable and can be utilized if it is preserved, published, and presented to end-users. The dissemination phase is concerned with the preservation and management of the information obtained from the preceding curation phases, as well as allowing users to access the data and benefit from the current curated content, and produce additional new information through manual individual or collaborative labeling if desired. Figure 5.1 shows the components of the data dissemination phase.

To enable data dissemination, a system is needed to facilitate the storage, use of and interaction with the data through searching for a desired set of images, exploring images, and adding new labels to the data. The requirements for such a system can be summarized as a) providing storage for the curated data, b) providing the ability to search and find images, and c) allowing users to further enrich the data by labeling and annotating images with relevant information.

For this purpose, we developed a simple collaborative in-house cloud-based infrastructure with a storage system that includes the curated data and a back-end and a front-end to allow image retrieval, browse, query, enrichment through labeling and annotation. The design and development of this platform was first motivated by the goal of providing support for a sensitive dataset, such as the human decomposition image dataset, that can not be disseminated using public systems due to privacy concerns. We refer to this platform as ICPUTRD: Image Cloud Platform for Use in labeling and Research on Decomposition.

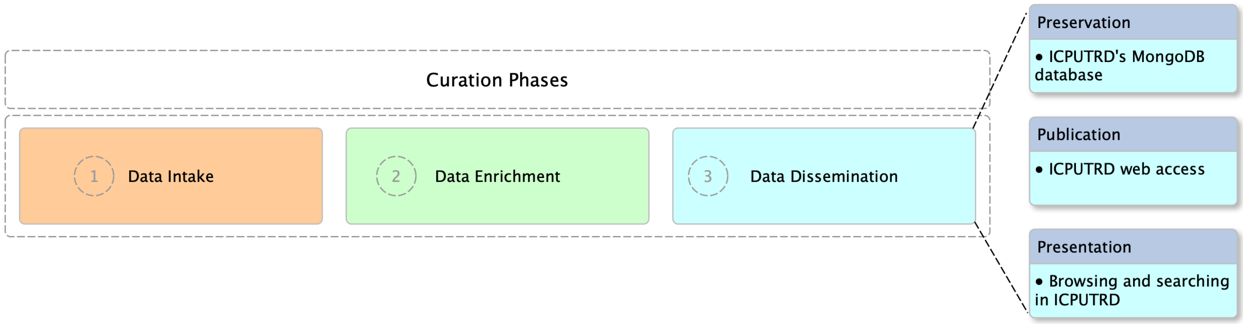


Figure 5.1: Data dissemination components.

Although ICPUTRD can be used for any dataset, it has been deployed for images of human decomposition in this work.

ICPUTRD is designed and implemented following the steps of the Software Development Life Cycle (SDLC) [114, 92] which includes requirements, design, implementation, and evaluation. The requirements for ICPUTRD are defined based on the features required to support data dissemination. Once the requirements are set, in the design step, a high-level design of the system and the decision on tools, techniques, and components needed for its implementation are made so that it can deliver each of the requirements. The system is then implemented following the choices made in the design step. Finally, ICPUTRD is evaluated via a user study to ensure its usability and deliverability of the requirements. Details on the requirements, design, implementation, and evaluation are provided in the following subsections.

5.1 Requirements

The requirements of ICPUTRD are set based on what is needed to enable data dissemination. These requirements are defined through iterative discussions with domain experts and categorized into three.

The first requirement concerns the ability to preserve the curated data for future use.

The second requirement is providing the ability to access the data. That means the system should provide a login system to allow eligible users can access the data.

The third requirement concerns the ability to use the curated data by providing effective search and interaction capabilities. The need for effective search is driven by the iterative process of building, refining, and evaluating hypotheses about the effects of various factors on a process such as decomposition. Therefore, the system should provide the ability to find images based on the structure of the data or various keywords/labels, representing the content of the images obtained from the previous curation phases.

Additionally, the system should allow users to further enrich the images with additional relevant information through labeling while allowing collaboration and quality control in the process. That means the system should provide users with a suitable interface that facilitates the enrichment process. In addition, the newly provided labels, along with the identity of their creator, should be stored for future reliable retrieval and quality control of them through providing features to view, edit, or delete them when needed.

5.2 Design and Implementation

For the system to meet the needs of the preservation requirement, it needs to have the curated data stored and have the ability to retrieve it when needed.

The preservation requirement can be addressed using a login system. The login system allows users can create accounts For the system to meet the needs of the search requirement, it should allow users to search for images based on the labels associated with them in the curation's enrichment phase. These labels are obtained from the metadata of the photos, external data sources, or provided by the domain expert and then extended to the entire dataset through our proposed auto-curation techniques.

In human decomposition data, these labels are the subject IDs, demographic information, the time at which each photo is taken, body parts, and decay features. ICPUTRD provides four different types of searching to fulfill the search requirement.

1. Search-by-subject allows users to request and obtain all or a subset of images associated with a particular subject.

2. Search-by-demographic-information, which allows users to search for all images associated with specific demographic attributes. Supported attributes are "age at death", "year of death", "sex", "ancestry" ("stature", "weight", and "date of placement".

3. Search-by-weather-history allows users to find images taken from subjects decomposing at a specific time of the year and exposed to certain levels of rain, humidity, temperature, and wind.

4. Search-by-image-content allows users to search for images depicting particular parts of the body or containing specific decay characteristics such as mummified or discolored skin, mold, maggots, or liquid purge.

In order to search for images based on subjects, demographic information, weather history, body parts, and decay features, a) the information should exist and be available, b) users should be provided with an interface that allows requesting for images based on such information, and c) the interface should be able to communicate with the stored data, extract and present it to users. Therefore, the search feature in ICPUTRD consists of three main components:

1. The client-side: a user interface that enables searching for desired content.
2. A database: it stores information so that it can be retrieved later.
3. The server-side: it handles incoming requests from the client-side and provides the logic for the communication between the client and the database.

These components are implemented using the MEAN stack which includes MongoDB [96] as the database system, ExpressJS [2] as the server-side JavaScript framework, AngularJS [1] as the front-end web framework for the client, and NodeJS [3] as the back-end run-time environment serving as the platform on which ExpressJS runs.

The user interface resides on the client-side within a web browser and can be accessed across the web by permitted users. It allows users to search, find, and view images. The interface is implemented using AngularJS, which is a JavaScript-based open-source web framework for developing user-facing applications.

The images in the human decomposition are high-resolution images. The image resolutions vary from 2400 × 1600 up to 4900 × 3200. The amount of storage needed to store the entire human decomposition dataset is approximately 4TB. Storing edited or augmented images increases the required storage space approximately eight-fold (32 TB of storage space). Searching for images and retrieving them from a database can be very slow and inefficient. Therefore, instead of storing all the images in a database, they are stored directly on the server's file system. Their paths, metadata, demographic information, weather information, body parts, and decay labels are stored in a database. ICPUTRD uses MongoDB as its storage system. Using a combination of MongoDB and the file system for image and metadata storage, the access process is drastically accelerated.

The server-side is where the logical operation and computation in response to user interactions take place, as is typically done in web-based applications. When users interact with the interface, commands are sent to the server-side based on user interactions (e.g., searching). After the necessary computations are done, the server-side responds with the appropriate information, which is then shown in the interface.

Figure 5.2 shows an overview of the communication between the client-side, database, and server-side. For example, when users search for all images of a subject/donor, the server-side receives this user action and communicates with the database to retrieve the requested information, finds the corresponding images from the file system, and displays them to the users through the web interface. ICPUTRD's server-side is implemented using Express.js. Express.js is a web application framework built using the JavaScript language. In addition, Figures 5.3, 5.4, and 5.5 show the user interface for the various types of search options provided by ICPUTRD. Users can search for images from specific donors/subjects and drill down to see individual images from that donor taken at specific dates (Figures 5.3, 5.4). Users can also search for images containing specific body parts of decay characteristics (Figure 5.5).

To fulfill the last requirement of the system, we needed to design and implement a labeling feature that allows users to further enrich the images with meaningful information. Such information can correspond to the entire image (image-level) or an area within the image (pixel-level). Similar to the search feature, the labeling process involves three client-side, database, and server-side components. The client-side allows the user to view existing labels or create new ones. The database stores the labels and associates them to their corresponding images. Finally, the server-side provides the logic for communication between the client and the database to store and retrieve labels. The labeling feature is implemented using the MEAN stack and the Annotorious library [4] that is used for annotating images.

The ICPUTRD platform provide the ability for labeling images at both image-level and pixel-level. In the image-level labeling interface, users are presented with a pool of images that can be labeled either one by one or multiple at the same time. An example of this interface is shown in Figure 4.3.

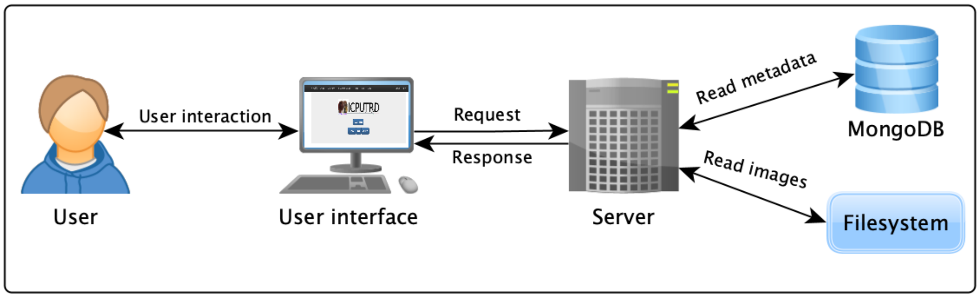
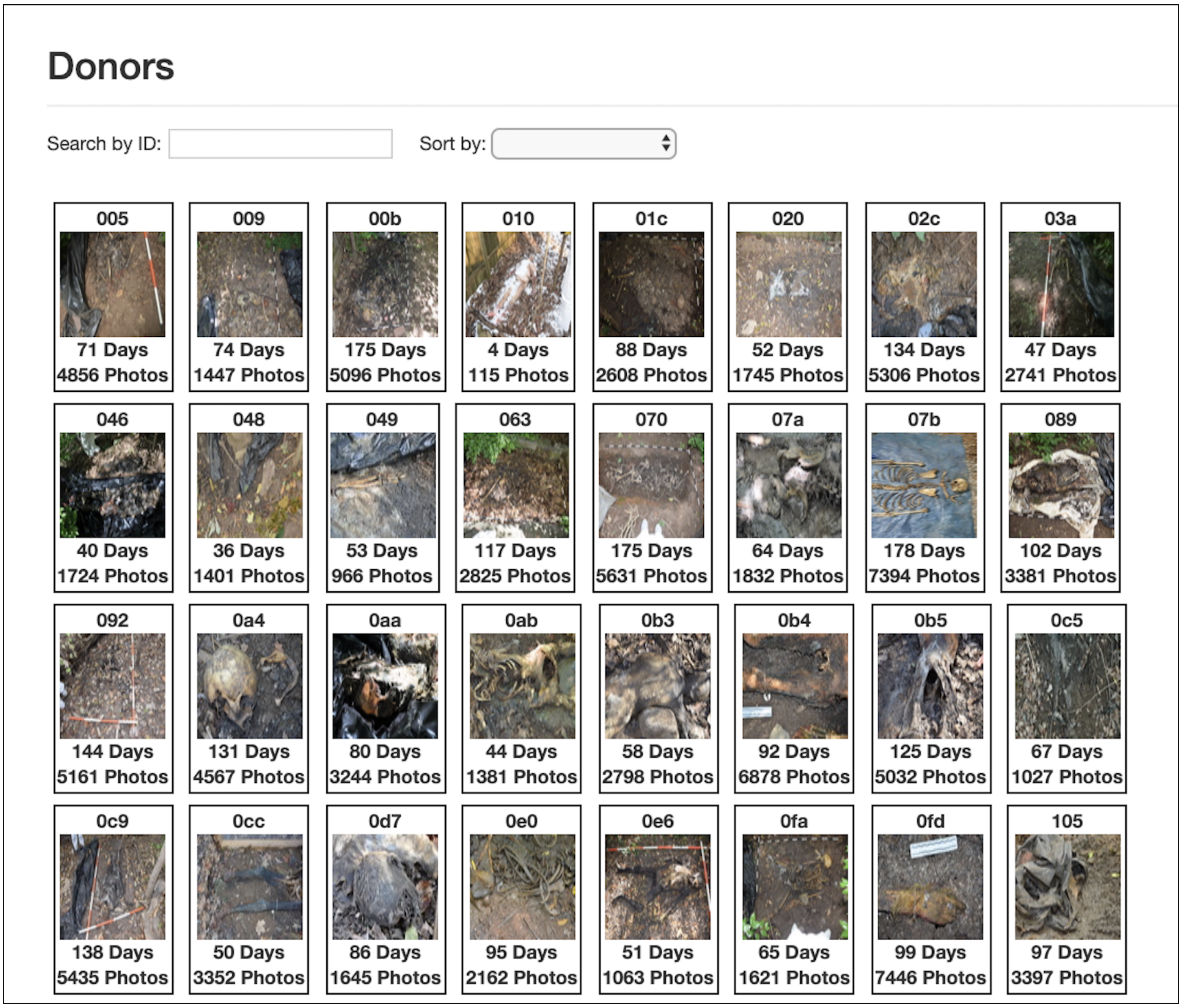


Figure 5.2: An overview of human interaction with ICPUTRD



Donors

Search by ID:

Sort by:

005
71 Days
4856 Photos

009
74 Days
1447 Photos

00b
175 Days
5096 Photos

010
4 Days
115 Photos

01c
88 Days
2608 Photos

020
52 Days
1745 Photos

02c
134 Days
5306 Photos

03a
47 Days
2741 Photos

046
40 Days
1724 Photos

048
36 Days
1401 Photos

049
53 Days
966 Photos

063
117 Days
2825 Photos

070
175 Days
5631 Photos

07a
64 Days
1832 Photos

07b
178 Days
7394 Photos

089
102 Days
3381 Photos

092
144 Days
5161 Photos

0a4
131 Days
4567 Photos

0aa
80 Days
3244 Photos

0ab
44 Days
1381 Photos

0b3
58 Days
2798 Photos

0b4
92 Days
6878 Photos

0b5
125 Days
5032 Photos

0c5
67 Days
1027 Photos

0c9
138 Days
5435 Photos

0cc
50 Days
3352 Photos

0d7
86 Days
1645 Photos

0e0
95 Days
2162 Photos

0e6
51 Days
1063 Photos

0fa
65 Days
1621 Photos

0fd
99 Days
7446 Photos

105
97 Days
3397 Photos

Figure 5.3: The search interface allows users to search for the donors/subjects based on their anonymized identification codes

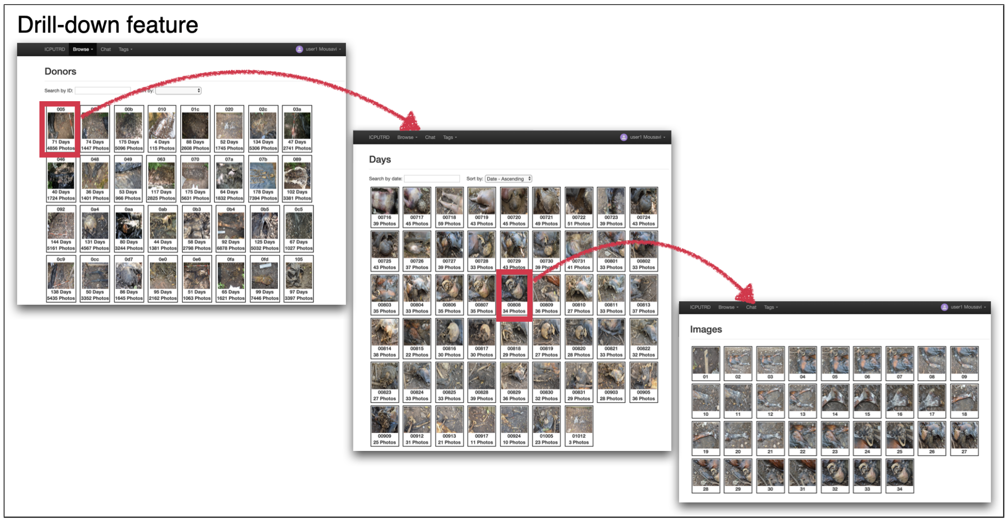
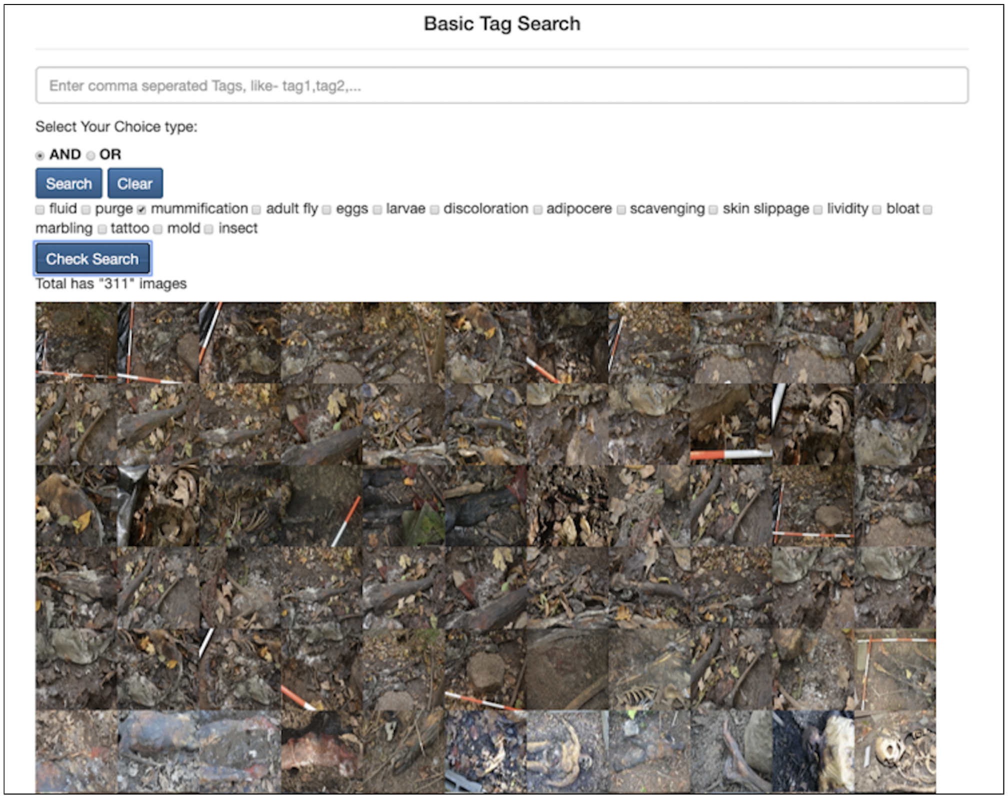


Figure 5.4: The drill-down feature in ICPUTRD that allows users to browse and explore images.



Basic Tag Search

Enter comma seperated Tags, like- tag1,tag2,...

Select Your Choice type:

● AND ○ OR

□ fluid □ purge ☑ mummification □ adult fly □ eggs □ larvae □ discoloration □ adipocere □ scavenging □ skin slippage □ lividity □ bloat □ marbling □ tattoo □ mold □ insect

Total has "311" images

Figure 5.5: An example of searching for specific labels in ICPUTRD.

In the pixel-level labeling interface, an image is shown to the user in full size through a web browser. When the interface loads the image, the server-side accesses the MongoDB database to look for any existing labels associated with the loaded image. The client-side draws the label(s) on the image if any is available. Users can then edit or remove existing labels and create new labels.

This functionality involves users drawing a polygon around the desired locations in an image by clicking. Each click creates a vertex connected to the next one until the user reaches the starting point and the polygon is closed. When the polygon is completely drawn, a pop-up dialog box appears so that the user can type the desired keyword for the selected area.

The developed nomenclature in subsection 3.1 is integrated into ICPUTRD to help users with the labeling process, ensure consistency across users as well as prevent spelling errors. As the user starts typing a keyword, ICPUTRD suggests potential matches from the developed nomenclature. The user then needs to only click on the desired keyword and hit the save button. The label information, which includes the keyword, coordinate of the drawn polygon, the user's ID, date of the annotation, and the image's name, will be stored in MongoDB to be later accessed when needed. Figure 5.6 shows an example of labeling an image.

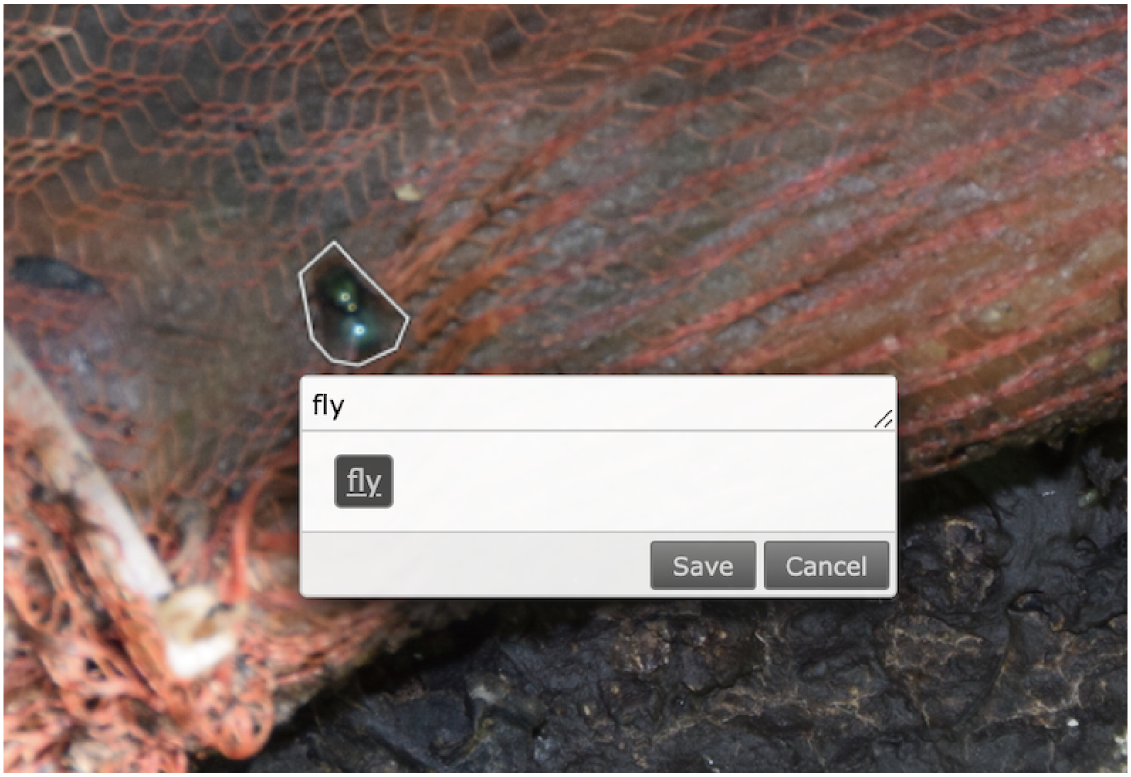
5.3 Evaluation and Results

In order to investigate whether ICPUTRD delivers the requirements for data dissemination, a system evaluation is conducted using an IRB-approved user study.

There are various ways to evaluate such a system. Nielsen defined four types of user-interface evaluation [102]:

1. Formal: using formulas to calculate usability measures
2. Informal: using the skill of the evaluator and their assessment of the system
3. Automatic: using a program to test the interface with various user specifications
4. Empirical: testing the interface by exposing it to the actual users

For our evaluation of ICPUTRD, we used the empirical approach, exposed the system to 27 users, and allowed them to interact with the system by performing a few predefined tasks.



fly

Figure 5.6: An example for the annotation interface provided in ICPUTRD.

The usability of a system can be measured using different approaches. Measurements often include three main aspects [12, 15]:

1. Effectiveness: users' ability to complete the tasks
2. Efficiency: the amount of resources used to do the tasks
3. Satisfaction: users' reaction to interacting with the system

These three aspects for ICPUTRD are assessed using an empirical approach which exposes ICPUTRD to the potential users through a user study containing specific tasks to interact with the system and a survey to learn about their experiences. Effectiveness, efficiency, and satisfaction are calculated by studying the result of the tasks and the survey.

We designed the user study tasks to evaluate the search and labeling abilities of ICPUTRD, its user-friendliness, its labeling interface, as well as the quality of the developed nomenclature. The tasks included finding images with relevant labels as well as labeling new images. The exact instruction and tasks for the study were as follows:

1. "Find an example image using the keywords (up to 5 keywords)". The main purpose of this task was to test the search capability provided by ICPUTRD.
2. "Now try this yourself! Type in a keyword, different from the five you just used, that you might want to use for research with this photo collection". This task was similar to the previous one but with a different goal. The main purpose of this task was to learn if any keywords were missing from the current nomenclature and to determine what keywords were frequently used by the participants.
3. "Using this image, label five keywords that you see (you can include new keywords)". For this task, all participants were shown the same image, and they were asked to label it. The purpose of this task was to determine if the users would be able to use the labeling capability of ICPUTRD.

To obtain information about the familiarity level of the participants with digital and forensic work and their experience from interacting with the system to perform the given tasks and the user study questions, we also designed pre- and post-study surveys.

The estimated time for each user to be introduced to the system and perform the tasks and the surveys was 30 minutes. The user study (including the surveys) was conducted while noting the participants' feedback.

The user study participants were potential users for digital human decomposition data: graduate students, professors, forensic anthropology staff, and law enforcement. Twenty-seven people, including one staff (MA degree), one professor (Ph.D. degree), 7 Ph.D. students, one master student, 11 osteology students from the Forensic Anthropology Center at the University of Tennessee, and 6 law enforcement officials participated in the user study.

In the surveys, the participants were asked about their familiarity level with digital and forensic work and if they were able to complete the tasks, and the difficulty they associated with each task. Figure 5.7 shows the familiarity level of the users from range 1 (not familiar) to 5 (very familiar). The plot indicates that most participants were fairly familiar with the decomposition concept and forensic work. However, they were not very experienced with digital databases and multimedia curation.

According to the post-study survey (5.8), the users were very comfortable with interacting with ICPUTRD even though they did not have prior experience in using such a system, which is an evidence of its efficiency and the satisfaction of the users [12, 15]. Figure 5.8 shows the convenience level of the participants with performing the three tasks (5: very easy, 1: very hard). According to the survey, 100%, 93% and 89% of the users finished tasks 1, 2, and 3 respectively. This result indicates the effectiveness [12] of ICPUTRD and the fact that participants were able to use the provided search and label features.

In addition, it was of interest to evaluate the adequacy of the nomenclature. To do so, we analyzed the keywords used by the participants for the second and third tasks. The labels used in the second task were mainly from the current nomenclature. However, the surveys show that only 18% of the users, mainly law enforcement users, were interested in finding images containing “gunshot wounds”, “knife wounds”, “rigor”, “bullae”, “blister”, “soldier fly”, “pupae”, or “foam” and could not find any. The reason for this was because of either the absence of such images in our dataset or missing these terms in our nomenclature. Such findings were used to improve the nomenclature.

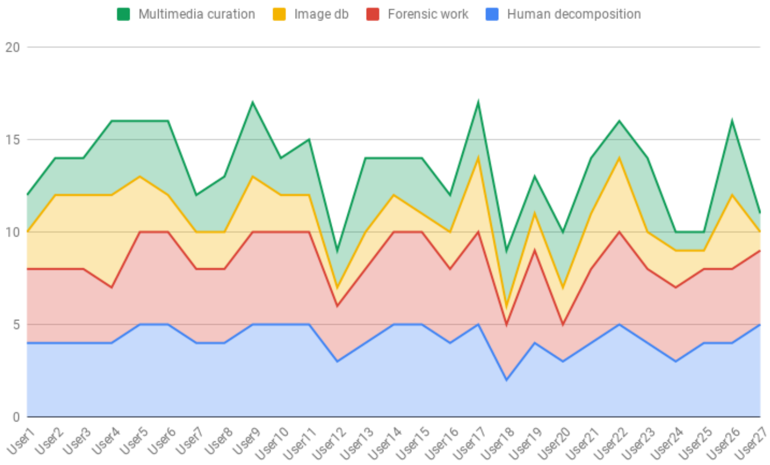


Figure 5.7: The familiarity level of the users from range 1 (not familiar) to 5 (very familiar) is shown.

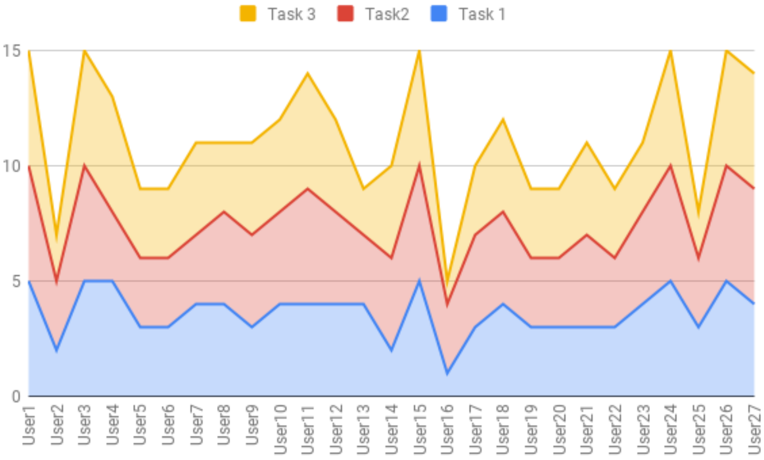


Figure 5.8: The convenience level of the participants with performing the three tasks (5: very easy, 1: very hard) is shown.

The labels created by 13 participants for the third task showed that same or similar keywords were being used. Among the 99 labels created by the participants, only 20 unique forensic labels were seen and others were repeated, indicating that participants were using the same terminology.

Overall, The results of the user study regarding the participants' experiences with ICPUTRD were very positive and confirm its suitability for data dissemination.

Chapter 6

Discussion and Conclusion

6.1 Goal of the Research

The goal of this research was to present an auto-curation framework that lays down the tasks needed to transform a collection of images from its raw format to an organized, contextualized, and enriched dataset ready to be used by end-users for various purposes, as well as algorithms, techniques, and tools that support performing these task.

This research was motivated by curating a large image collection of human decomposition to support statistical analysis and research in the forensic anthropology domain, which can shift the current fieldwork-based norm in research in this domain toward digital-based research. With the availability of a large well curated image dataset in this domain, a) studying the decomposition process will no longer be limited to only the period of decomposition itself, b) it will allow researchers to focus on particular factors and correlate them to environmental and individual characteristics (e.g., age or weight), and c) it will enable testing hypotheses and studying the decomposition process using a large sample size of subjects.

6.2 Key Outcomes

In this dissertation we have developed an auto-curation framework along with designing and developing tools and techniques to support the curation tasks for large image datasets with evolving content. Specifically

- We defined curation tasks for large image datasets and implemented these tasks in the context of image datasets representing evolving content

- Utilized the characteristics of evolving datasets to design and develop unsupervised techniques for performing the organization task of the curation without human involvement

- Developed a human-machine collaborative technique to minimize the curator's efforts in the curation process and simplify mass labeling

- Designed and developed an ML-based technique to reuse and propagate the limited number of expert-provided labels to a large portion of the dataset by utilizing the characteristics of evolving image data

- Enabled future reusability and further enrichment of the curated content by providing a platform to support collaborative and crowdsourced efforts to iteratively refine the curated content as well as preserve, present and publish the curated data

6.3 Discussion

Image datasets with evolving content, such as the human decomposition images, have unique characteristics that make them different than other image datasets. Some examples are, 1) subjects gradually change/evolve through time, causing drastic differences between the appearance of images depicting the same content, 2) subjects do not experience the same level and type of evolution, 3) images of the same subject residing on a different similarity space compared to images between different subjects (there are different dimensions to the similarity in evolving data), 4) images of the same subject, even-though share very little similarity in their appearances, can have similar annotations, 5) the evolution might be tracked with varying intervals between images of a subject. These characteristics result in both challenges and opportunities in using machine learning algorithms to automate the curation process of such datasets.

Challenges are due to the non-uniform level and type of evolution for the subjects, multiple dimensions of similarity, and the diminution of similarity between images depicting the same content that confounds clustering and other machine learning approaches that rely on image similarity measures. However, the gradual changes in such data result in a local similarity between images of the same subjects in neighboring timesteps in the evolution timeline, increasing the likelihood of having images with very similar annotations, which in turn creates opportunities for providing more training data for machine learning methods and improving their performance.

6.4 Conclusion

Organizing, contextualizing, and enriching a large image dataset, or in other words curating data, is a time-consuming and expensive task, yet it is necessary to be performed on a dataset to reach its full potential. As a result, in this work, we first defined curation for image datasets as the process of turning unstructured or semi-structured image collections into unified, cleaned, organized, contextualized, and enriched datasets ready to be utilized by end-users and researchers for various applications. We then introduced a framework for auto-curating large image datasets that included three main overarching phases that encompass this definition: intake, enrichment, and dissemination. Subsequently, we presented AI-assisted techniques and tools to support performing curation without human intervention, assisting experts in the curation process when their intervention is required and automatically expanding on their provided input to reduce their effort and accelerate the curation process. The proposed auto-curation framework was then employed to curate a large image dataset tracking human decomposition that is of great value for the forensic anthropology researchers, as well as tested the approaches using other smaller evolving image collections.

Our work in this dissertation has several implications. First, it leads to producing curated data semi-automatically, which can enable image retrieval, browsing, querying, etc. in large image collections. Second, it facilitates producing curated data for niche domains and opens the path to research and analysis for new applications in these domains which lack curated data and labeling can be costly. For example, in the forensic anthropology domain, the availability of a large, curated dataset allows research and analysis on one of the most important topics in this domain: postmortem interval estimation. Finally, this work enables studying and analyzing co-occurrence, speed, progression, and prediction of features in evolving data.

Bibliography

[1] (accessed Aug 2020). angular. https://angular.io/. 96

[2] (accessed Aug 2020). expressjs. https://expressjs.com/. 96

[3] (accessed Aug 2020). nodejs. https://nodejs.org/en/. 96

[4] (accessed July 20. 2020). annotorious. https://www.ait.ac.at/en/research-topics/data-science/prototypes-demos/annotorious-image-annotation/. 97

[5] Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003). A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 81–92. VLDB Endowment. 19, 39

[6] Andriluka, M., Uijlings, J. R., and Ferrari, V. (2018). Fluid annotation: a human-machine collaboration interface for full image annotation. *arXiv preprint arXiv:1806.07527*. 20, 24, 132, 133

[7] Bae, E. and Bailey, J. (2006). Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 53–62. IEEE. 18

[8] Bautista, M. A., Sanakoyeu, A., Tikhoncheva, E., and Ommer, B. (2016). Cliquecnn: Deep unsupervised exemplar learning. In *Advances in Neural Information Processing Systems*, pages 3846–3854. 39

[9] Bell, J. and Dee, H. M. (2016). Aberystwyth leaf evaluation dataset. 14, 84

[10] Benjdira, B., Bazi, Y., Koubaa, A., and Ouni, K. (2019). Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images. *Remote Sensing*, 11(11):1369. 21

[11] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. (2019). Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*. 70

[12] Bevan, N. and Macleod, M. (1994). Usability measurement in context. *Behaviour & information technology*, 13(1-2):132–145. 102, 103

[13] Bilen, H. and Vedaldi, A. (2016). Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854. 133

[14] Boef, A. G., Dekkers, O. M., Vandenbroucke, J. P., and le Cessie, S. (2014). Sample size importantly limits the usefulness of instrumental variable methods, depending on instrument strength and level of confounding. *Journal of clinical epidemiology*, 67(11):1258–1264. 72

[15] Brooke, J. (1996). Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189. 102, 103

[16] Buchner, J. (2020). Imagehash. `https://github.com/JohannesBuchner/imagehash`. 37

[17] Caesar, H., Uijlings, J., and Ferrari, V. (2016). Coco-stuff: Thing and stuff classes in context. *CoRR, abs/1612.03716*, 5:8. 20, 22, 132, 133

[18] Cantürk, İ. and Özyılmaz, L. (2018). A computational approach to estimate postmortem interval using opacity development of eye for human subjects. *Computers in biology and medicine*, 98:93–99. 72

[19] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. In *Face and Gesture Recognition*. 2, 16

[20] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149. 18, 39, 41, 43, 49, 50, 77

[21] Caruana, R., Elhawary, M., Nguyen, N., and Smith, C. (2006). Meta clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 107–118. IEEE. 18

[22] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution,

and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848. 83

[23] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848. 20, 133

[24] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258. 83

[25] Constantopoulos, P. and Dallas, C. (2008). Aspects of a digital curation agenda for cultural heritage. In *2008 IEEE International Conference on Distributed Human-Machine Systems. Athens, Greece: IEEE*, pages 1–6. 5

[26] Constantopoulos, P., Dallas, C., Androutsopoulos, I., Angelis, S., Deligiannakis, A., Gavrilis, D., Kotidis, Y., and Papatheodorou, C. (2009). Dcc&u: An extended digital curation lifecycle model. *International Journal of Digital Curation*, 4(1). 16

[27] Contardo, G., Denoyer, L., and Artières, T. (2017). A meta-learning approach to one-step active learning. *arXiv preprint arXiv:1706.08334*. 20

[28] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 20, 22, 24, 70, 132, 133

[29] Corti, L., Van den Eynden, V., Bishop, L., and Woollard, M. (2019). *Managing and sharing research data: a guide to good practice*. SAGE Publications Limited. 16

[30] Cox, A. M. and Tam, W. W. T. (2018). A critical analysis of lifecycle models of the research process and research data management. *Aslib Journal of Information Management*. 16

[31] Cragin, M. H., Heidorn, P. B., Palmer, C. L., and Smith, L. C. (2007). An educational program on data curation. 4, 16

[32] Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Complex Systems:1695. 138

[33] Dang, X. H. and Bailey, J. (2010). Generation of alternative clusterings using the cami approach. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 118–129. SIAM. 18

[34] De Brabandere, B., Neven, D., and Van Gool, L. (2017). Semantic instance segmentation for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–9. 21

[35] De Hoon, M. J., Imoto, S., Nolan, J., and Miyano, S. (2004). Open source clustering software. *Bioinformatics*, 20(9):1453–1454. 38

[36] Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., et al. (2014). Feedback on a publicly distributed image database: the messidor database. *Image Analysis & Stereology*, 33(3):231–234. 23

[37] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. 20, 32, 38, 43, 61, 76, 77, 129, 133

[38] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142. 38

[39] Deselaers, T., Alexe, B., and Ferrari, V. (2010). Localizing objects while learning their appearance. In *European conference on computer vision*, pages 452–466. Springer. 133

[40] Dópido, I., Li, J., Marpu, P. R., Plaza, A., Dias, J. M. B., and Benediktsson, J. A. (2013). Semisupervised self-learning for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, 51(7):4032–4044. 22

[41] Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774. 39

[42] Dwyer, J. L., Roy, D. P., Sauer, B., Jenkerson, C. B., Zhang, H. K., and Lymburner, L. (2018). Analysis ready data: enabling analysis of the landsat archive. *Remote Sensing*, 10(9):1363. 17

[43] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2007). The pascal visual object classes challenge 2007 (voc2007) results. 38, 70

[44] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338. 8, 23, 84

[45] Fang, H.-S., Lu, G., Fang, X., Xie, J., Tai, Y.-W., and Lu, C. (2018). Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. *arXiv preprint arXiv:1805.04310*. 69

[46] Fang, M., Li, Y., and Cohn, T. (2017). Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383*. 20

[47] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88. 5

[48] Forsyth, D. A. and Ponce, J. (2012). *Computer vision: a modern approach*. Pearson,. 22

[49] Freitas, A. and Curry, E. (2016). *Big Data Curation*, pages 87–118. Springer International Publishing, Cham. 5, 16

[50] Fukui, T. and Wada, T. (2014). Commonality preserving image-set clustering based on diverse density. In *International Symposium on Visual Computing*, pages 258–269. Springer. 38

[51] Galloway, A., Birkby, W. H., Jones, A. M., Henry, T. E., and Parks, B. O. (1989). Decay rates of human remains in an arid environment. *Journal of Forensic Science*, 34(3):607–616. 72

[52] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. 22

[53] Gelderman, H., Kruiver, C., Oostra, R., Zeegers, M., and Duijst, W. (2019). Estimation of the postmortem interval based on the human decomposition process. *Journal of Forensic and Legal Medicine*, 61:122–127. 72

[54] Gokberk Cinbis, R., Verbeek, J., and Schmid, C. (2014). Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2409–2416. 21, 133

[55] Grimaudo, L., Mellia, M., Baralis, E., and Keralapura, R. (2014). Select: Self-learning classifier for internet traffic. *IEEE Transactions on Network and Service Management*, 11(2):144–157. 22

[56] Guérin, J., Gibaru, O., Thiery, S., and Nyiri, E. (2017). Cnn features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700*. 17, 18, 38, 39, 41, 49

[57] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE. 20, 133

[58] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 32, 43, 77, 83, 129

[59] Huang, L., Chao, H.-Y., and Wang, C.-D. (2019). Multi-view intact space clustering. *Pattern Recognition*, 86:344–353. 18

[60] Hung, W.-C., Tsai, Y.-H., Liou, Y.-T., Lin, Y.-Y., and Yang, M.-H. (2018). Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*. 23, 83

[61] Hyde, R., Angelov, P., and MacKenzie, A. R. (2017). Fully online clustering of evolving data streams into arbitrarily shaped clusters. *Information Sciences*, 382:96–114. 19

[62] Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., and Schiele, B. (2017). Arttrack: Articulated multi-person tracking in the wild. In *CVPR'17*. 128

[63] Jain, P., Meka, R., and Dhillon, I. S. (2008). Simultaneous unsupervised learning of disparate clusterings. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 1(3):195–210. 18

[64] Johnston, L. R., Carlson, J., Hudson-Vitale, C., Imker, H., Kozlowski, W., Olendorf, R., Stewart, C., Blake, M., Herndon, J., McGeary, T. M., et al. (2018). Data curation network: A cross-institutional staffing model for curating research data. 16

[65] Kalluri, T., Varma, G., Chandraker, M., and Jawahar, C. V. (2019). Universal semi-supervised semantic segmentation. 22

[66] Ke, Z., Di Qiu, K. L., Yan, Q., and Lau, R. W. (2020). Guided collaborative training for pixel-wise semi-supervised learning. In *ECCV*, volume 2, page 6. Springer. 70

[67] Ketkar, N. (2017). Stochastic gradient descent. In *Deep learning with Python*, pages 113–132. Springer. 83

[68] Khoreva, A., Benenson, R., Hosang, J. H., Hein, M., and Schiele, B. (2017). Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, volume 1, page 3. 21, 133

[69] Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. (2019). Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9404–9413. 21

[70] Kolesnikov, A. and Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711. Springer. 133

[71] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., et al. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*. 20, 133

[72] Laine, S. and Aila, T. (2017). Temporal ensembling for semi-supervised learning. 22

[73] Laine, S. M. and Aila, T. O. (2018). Temporal ensembling for semi-supervised learning. US Patent App. 15/721,433. 70

[74] Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3. 70

[75] Lee, H.-W., Kim, N.-r., and Lee, J.-H. (2017). Deep neural network self-training based on unsupervised learning and dropout. *International Journal of Fuzzy Logic and Intelligent Systems*, 17(1):1–9. 22

[76] Lee, J., Kim, E., Lee, S., Lee, J., and Yoon, S. (2019). Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5267–5276. 69

[77] Li, Q., Arnab, A., and Torr, P. H. (2018). Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 102–118. 21

[78] Li, Y., Chen, X., Zhu, Z., Xie, L., Huang, G., Du, D., and Wang, X. (2019). Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7026–7035. 21

[79] Liao, R., Schwing, A., Zemel, R., and Urtasun, R. (2016). Learning deep parsimonious representations. In *Advances in Neural Information Processing Systems*, pages 5076–5084. 39

[80] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context. 22

[81] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer. 22, 23, 69

[82] Liu, H., Shao, M., Li, S., and Fu, Y. (2016a). Infinite ensemble for image clustering. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1745–1754. 38

[83] Liu, R., Palsetia, D., Paul, A., Al-Bahrani, R., Jha, D., Liao, W.-k., Agrawal, A., and Choudhary, A. (2016b). Pinternet: A thematic label curation tool for large image datasets. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2353–2362. IEEE. 17

[84] Lord, P., Macdonald, A., Lyon, L., and Giaretta, D. (2004). From data deluge to data curation. In *Proceedings of the UK e-science All Hands meeting*, pages 371–375. Citeseer. 16

[85] Ma, Y., Liu, Y., Xie, Q., and Li, L. (2019). Cnn-feature based automatic image annotation method. *Multimedia Tools and Applications*, 78(3):3767–3780. 21

[86] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605. 129

[87] Malik, H. H. and Bhardwaj, V. S. (2011). Automatic training data cleaning for text classification. In *2011 IEEE 11th international conference on data mining workshops*, pages 442–449. IEEE. 35

[88] Maninis, K.-K., Caelles, S., Pont-Tuset, J., and Van Gool, L. (2018). Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625. 21, 133

[89] Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103. 41, 48

[90] McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics. 22

[91] Megyesi, M. S., Nawrocki, S. P., and Haskell, N. H. (2005). Using accumulated degree-days to estimate the postmortem interval from decomposed human remains. *Journal of Forensic Science*, 50(3):1–9. 129

[92] Mishra, A. and Dubey, D. (2013). A comparative study of different software development life cycle models in different scenarios. *International Journal of Advance research in computer science and management studies*, 1(5). 94

[93] Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993. 70

[94] Mockus, A. (2014). Engineering big data solutions. In *Future of Software Engineering Proceedings*, pages 85–99. 1, 2

[95] Molino, P., Wang, Y., and Zhang, J. (2019). Parallax: Visualizing and understanding the semantics of embedding spaces via algebraic formulae. *arXiv preprint arXiv:1905.12099*. 32

[96] MongoDB, I. (Accessed 2020). mongodb. `https://www.mongodb.com/`. 96

[97] Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R., and Yuille, A. (2014). The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898. 20, 24, 133

[98] Mousavi, S., Lee, D., Griffin, T., Cross, K., Steadman, D., and Mockus, A. (2021). Schism: Semantic clustering via image sequence merging for images of human-decomposition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2190–2199. 19, 41, 74

[99] Mousavi, S., Lee, D., Griffin, T., Steadman, D., and Mockus, A. (2019a). An analytical workflow for clustering forensic images. *arXiv preprint arXiv:2001.05845*. 17, 18, 128

[100] Mousavi, S., Lee, D., Griffin, T., Steadman, D., and Mockus, A. (2020). Collaborative learning of semi-supervised clustering and classification for labeling uncurated data. *arXiv preprint arXiv:2003.04261*. 39, 58

[101] Mousavi, S., Nabati, R., Kleeschulte, M., Steadman, D., and Mockus, A. (2019b). Machine-assisted annotation of forensic imagery. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1595–1599. IEEE. 39, 134

[102] Nielsen, J. (1994). Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. 100

[103] Niu, D., Dy, J. G., and Jordan, M. I. (2010). Multiple non-redundant spectral clustering views. 18

[104] Ouali, Y., Hudelot, C., and Tami, M. (2020). Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684. 23, 74, 83, 84, 88

[105] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. 83

[106] Pathak, D., Krahenbuhl, P., and Darrell, T. (2015). Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804. 133

[107] Philbrick, K. A., Weston, A. D., Akkus, Z., Kline, T. L., Korfiatis, P., Sakinis, T., Kostandy, P., Boonrod, A., Zeinoddini, A., Takahashi, N., et al. (2019). Ril-contour: a medical imaging dataset annotation tool for and with deep learning. *Journal of digital imaging*, pages 1–11. 21

[108] Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabuddhe, V., and Meriaudeau, F. (2018). Indian diabetic retinopathy image dataset (idrid): a database for diabetic retinopathy screening research. *Data*, 3(3):25. 23

[109] Raevich, A., Dobronets, B., Popova, O., and Raevich, K. (2020). Conceptual model of operational–analytical data marts for big data processing. In *E3S Web of Conferences*, volume 149, page 02011. EDP Sciences. 1

[110] Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271. 20, 133

[111] Ricanek, K. and Tesafaye, T. (2006). Morph: A longitudinal image database of normal adult age-progression. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 341–345. IEEE. 13, 41

[112] Ringwald, T. and Stiefelhagen, R. (2021). Adaptiope: A modern benchmark for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 101–110. 35

[113] Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. *WACV/MOTION*, 2. 22

[114] Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3):8–13. 94

[115] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173. 20, 24, 133

[116] Schaefer, G. and Stich, M. (2003). Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472–480. International Society for Optics and Photonics. 38

[117] Sener, O. and Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*. 20

[118] Shamma, D. A. (Retrieved 4/24/2021.). One hundred million creative commons flickr images for research. In *Yahoo Reaserch*. 2, 16

[119] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813. 39

[120] Sharma, N. and Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *Journal of medical physics/Association of Medical Physicists of India*, 35(1):3. 21

[121] Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., and Anandkumar, A. (2017). Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*. 20

[122] Simmons, T., Adlam, R. E., and Moffatt, C. (2010). Debugging decomposition data—comparative taphonomic studies and the influence of insects and carcass size on decomposition rate. *Journal of forensic sciences*, 55(1):8–13. 72

[123] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 77, 138

[124] Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*. 70

[125] Souly, N., Spampinato, C., and Shah, M. (2017). Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE international conference on computer vision*, pages 5688–5696. 23, 83

[126] Stonebraker, M., Bruckner, D., Ilyas, I. F., Beskales, G., Cherniack, M., Zdonik, S. B., Pagan, A., and Xu, S. (2013). Data curation at scale: the data tamer system. In *Cidr*. 16

[127] Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., and Wang, J. (2019). High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*. 83

[128] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9. 43, 77

[129] Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204. 70

[130] Thirumuruganathan, S., Tang, N., Ouzzani, M., and Doan, A. (2018). Data curation with deep learning [vision]. *arXiv preprint arXiv:1803.01384*. 35

[131] Tian, F., Gao, B., Cui, Q., Chen, E., and Liu, T.-Y. (2014). Learning deep representations for graph clustering. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. 18

[132] Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schuberth, P., Mayr, A., Heusel, M., Hofmarcher, M., Widrich, M., et al. (2016). Speeding up semantic segmentation for autonomous driving. In *MLITS, NIPS Workshop*, volume 2. 21

[133] Wang, R. Y. and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33. 35

[134] Wang, X., Lu, L., Shin, H.-C., Kim, L., Bagheri, M., Nogues, I., Yao, J., and Summers, R. M. (2017). Unsupervised joint mining of deep features and image labels for large-scale

radiology image categorization and scene recognition. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 998–1007. IEEE. 17

[135] Wang, Z., Chang, S., Zhou, J., Wang, M., and Huang, T. S. (2016). Learning a task-specific deep architecture for clustering. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 369–377. SIAM. 18

[136] Weston, J., Ratle, F., Mobahi, H., and Collobert, R. (2012). Deep learning via semi-supervised embedding. In *Neural networks: Tricks of the trade*, pages 639–655. Springer. 22

[137] Wissik, T. and Ďurčo, M. (2016). Research data workflows: from research data lifecycle models to institutional solutions. In *Selected Papers from the CLARIN Annual Conference 2015, October 14–16, 2015, Wroclaw, Poland*, number 123, pages 94–107. Linköping University Electronic Press. 16

[138] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52. 43, 77, 129

[139] Wulder, M. A., White, J. C., Loveland, T. R., Woodcock, C. E., Belward, A. S., Cohen, W. B., Fosnight, E. A., Shaw, J., Masek, J. G., and Roy, D. P. (2016). The global landsat archive: Status, consolidation, and direction. *Remote Sensing of Environment*, 185:271–283. 16

[140] Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. (2016). Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22. 20, 24, 133

[141] Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. 39

[142] Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., and Urtasun, R. (2019). Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8818–8826. 21

[143] Xiong, Z., Wang, Z., Du, C., Zhu, R., Xiao, J., and Lu, T. (2018). An asian face dataset and how race influences face recognition. In *Pacific Rim Conference on Multimedia*, pages 372–383. Springer. 2, 16

[144] Xu, Y.-M., Wang, C.-D., and Lai, J.-H. (2016). Weighted multi-view clustering with feature selection. *Pattern Recognition*, 53:25–35. 18

[145] Yang, J., Parikh, D., and Batra, D. (2016). Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156. 17, 18, 39

[146] Yang, S. and Zhang, L. (2017). Non-redundant multiple clustering by nonnegative matrix factorization. *Machine Learning*, 106(5):695–712. 18

[147] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196. 22

[148] Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*. 21

[149] Zhang, K., Albiero, V., and Bowyer, K. W. (2020). A method for curation of web-scraped face image datasets. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE. 17, 35

[150] Zhou, A., Cao, F., Qian, W., and Jin, C. (2008). Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems*, 15(2):181–214. 19

[151] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 4. IEEE. 20, 24, 133

[152] Zhou, Y., He, X., Huang, L., Liu, L., Zhu, F., Cui, S., and Shao, L. (2019). Collaborative learning of semi-supervised segmentation and classification for medical

images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2079–2088. 23

[153] Zhu, J.-J. and Bento, J. (2017). Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*. 20

[154] Zou, Y., Zhang, Z., Zhang, H., Li, C.-L., Bian, X., Huang, J.-B., and Pfister, T. (2020). Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*. 23, 74, 84, 86, 88

Appendices

In this section, I have provided two of my works related to the human decomposition dataset that are not included in the main body of my dissertation but can provide insight and ideas for similar applications.

A An Analytical Workflow for Clustering Forensic Images

A.1 Introduction

Images that show stages of human decomposition, represent high potential value to forensic research and law enforcement. The main sources for such images are forensic anthropology centers and crime scenes. Applications and users benefit from these image collections when labeled with relevant forensic classes, thus improving querying images with the desired content.

Our dataset collected at the University of Tennessee's Anthropology Research Center contains 1 million images collected over 8 years. Manual labeling is infeasible due to the time and effort required given the sheer number and heterogeneity, different camera angles, body parts and decomposition rates, of the dataset. Additionally, creating enough labels for successful supervised learning is also difficult due the scarcity of forensic experts, and the graphic nature of the images.

Human part detection methods [62] do not perform well because of the deformations and decay of the bodies due to environmental factors as well as the natural decay over time. An example image from a dataset containing human decomposition is shown in Figure 1.

In this work, we present an unsupervised analytical workflow [99][1], shown in Figure 2, for clustering forensic images. We have found that the key variation in features in such image collections resides along two dimensions: a) different body parts represented in the image and b) different stages of decomposition. Our approach combines information representing

[1] Mousavi, Sara, et al. "An Analytical Workflow for Clustering Forensic Images (Student Abstract)." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 10. 2020.

domain knowledge about decomposition with features extracted from the image content to group images along both dimensions.

First, image features using ResNet50 [58] pre-trained on ImageNet [37] are extracted. However, per our experiments, ResNet does not capture decomposition-related aspects that are absent from ImageNet. We, therefore, incorporate decomposition-related metadata such as temperature, humidity, and wind speed using Accumulated Degree Days (ADD) [91]. Next, we use t-SNE [86] to get a sense of the number of potential clusters in the data and then cluster the features. Finally we manually evaluate our clustering method using a web interface we designed.

Using our method, we were able to cluster 8507 images into 15 clusters with an average precision of 89%.

A.2 Method

The developed workflow is shown in Figure 2 We exclude the top layers of a pre-trained ResNet50 model since the classes of our collection are completely different. The model is pre-trained on ImageNet [37] and produces a 2048 length feature vector for each image which is then reduced to 256 via PCA [138]. Our initial approach consisted of directly clustering the feature vectors afterwards. This resulted in clusters that neither were separated by the body parts nor decomposition stages. To improve on this, we extended the feature vectors with external metadata. In our dataset, the only metadata available for each photo is an anonymous *Id* of the donor and the date the photograph was taken. We use this information in combination with external weather data we obtained for geographic location of the body farm and time of the photos. Hourly temperature, humidity, and wind speed are used to calculate accumulated degree-days, ADD, for temperature, humidity and wind speed. ADD is commonly used to estimate the postmortem interval [91] in Forensic Anthropology.

In order to include the weather information in our clustering, we generate a numeric vector, W, with the size of 3. Values in W are ADDs for temperature, humidity and wind speed. In this case, ADD for temperature is calculated as $\frac{T_{d1}+T_{d2}+\cdots+T_{dn}}{n}$, where T_{di} is the average temperature for the ith day, and n is the total number of the days since decomposition. ADD for humidity and wind speed are calculated in a similar manner. A

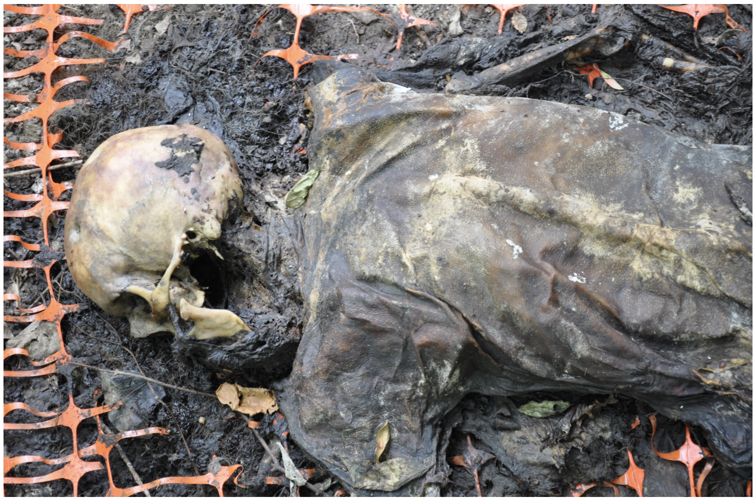


Figure 1: An example image of a decaying body after a month of being exposed to summer weather.

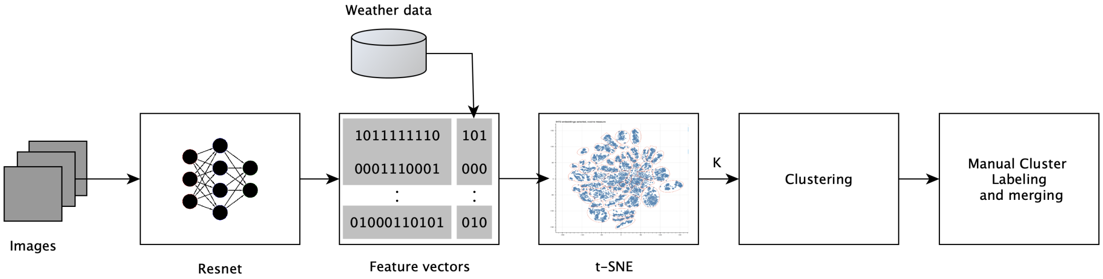


Figure 2: The overall architecture of our workflow is shown. Images are fed into a ResNet model to obtain feature vectors. The features are then combined with weather data. Using t-SNE we find the potential number of clusters in the data and then perform clustering using KMeans. The resulting clusters are then manually labeled and merged. The workflow is implemented using Keras, Python3, HTML, and Javascript. We used a Quadro M6000 GPU for generating the ResNet feature vectors.

new representation for each photo, P_i, is created by appending the corresponding W to the current vector representation.

Most clustering techniques require an estimate of the number of clusters. We visualized the generated $256 + 3 = 259$ length vectors for the photos in 2D using t-SNE to find the potential number of clusters in feature-space (included in Figure 2). Based on this plot we chose 50 clusters and use KMeans clustering technique.

To evaluate the technique we built a web interface supporting browsing through the images in each cluster. We then selected the miss-clustered images by simply clicking on them. Each selection appends the image name to a text file that can be downloaded at the end of evaluation. Counting the number of the miss-classified images, we calculated the precision of the clustering method. The web interface also allows labeling the clusters with a meaningful keyword. The goal of this cluster-level labeling exercise was to group images of the same body part and order them early to late stages of decomposition based on time.

A.3 Results and Conclusion

In order to test our clustering method, we started with 8507 photos taken from two donors over 127 and 122 photography sessions within 8 months.Merging the initial 50 clusters from the described data resulted in 12 clusters.

Our findings show that by adding weather features, the clustering precision increased to 89%, from the initial approach that yielded only 64%.

In conclusion, we developed an analytical workflow that incorporates external metadata with the image feature representations to cluster a large temporal forensic dataset in an unsupervised manner. The resulting clusters not only provide a structured way for the users to navigate a large image collection, but also paves the path for providing data for supervised classification, object detection, and semantic segmentation.

B Machine-Assisted Annotation of Forensic Imagery

B.1 Introduction

Certain image collections, such as images of human decomposition, represent high potential value to forensic research and law enforcement, yet are scarce, have restricted access, and are very difficult to utilize. To utilize such image collections, they need to be annotated with relevant forensic classes so that a user can find images with the desired content. This work is motivated by our attempt to annotate over one million photos taken over seven years in a facility focused on studying human decomposition.

Annotating images is a difficult task in general, with a single image taking from 19 minutes [17] to 1.5 hours [28] on average. Human decomposition images present additional difficulties. First, forensic data cannot be crowd-sourced due to its graphic nature and need for anonymity. Second, annotating forensic classes requires experts in human decomposition that are hard to come by. Therefore, it is natural to consider approaches to support such manual effort with machine learning (ML) techniques [6]. Unique challenges specific to forensic images prevent direct application of state-of-the art techniques described in, for example, [6]. This is mainly due to the primary focus of the annotation in being used by researchers, not algorithms. Particularly, when it comes to creating relevant training samples for ML approaches, we encountered the following challenges and discuss them afterwards:

- It is not feasible to annotate images completely. In other words, the user may choose to only annotate some instances of a class in an image, or only a subset of classes.
- The locations of forensically-relevant objects is not precisely outlined but, instead, roughly indicated via rectangular areas.
- It is not feasible to annotate a very large number of examples of a forensic class.

The first challenge results from the numerous instances of certain classes (for example, there may be tens of thousands of maggots in a single image, spread in multiple groups). Annotators may only tag classes relevant to their investigation or classes that they have sufficient expertise to identify accurately. The second challenge is caused by the primary objective of the annotator to provide indicators to other researchers and the need to maximize the number of manually annotated images irrespective of the ability of machine learning to generalize from them (i.e. using simple rectangles instead of more time-consuming masks). The last challenge is imposed by the limited availability of forensic experts. Furthermore, since it is not possible to annotate the entire set of images, the expert needs to choose which images to annotate. Choosing images randomly, as it turns out, is highly inefficient since such images rarely contain relevant forensic classes.

LabelMe [115] and similar polygon-drawing interfaces have been used to annotate image collections [28, 97, 140, 151]. The annotators need to manually select the areas of interest and label them with the correct label. Given the amount of time needed to annotate a single image, such approaches are not suitable for annotating one million forensic images.

Fluid Annotation [6] assists annotators in fully annotating images by providing initial annotations, that can be edited as needed. Fluid annotation uses Mask-RCNN [57] as the primary deep learning model. For Mask-RCNN and other deep-learning based techniques such as Deeplabv3+ and YOLO [23, 110] to work, large, complete, and clean training datasets such as Open Images, Image Net and COCO [71, 37, 17] are required. Such approaches without additional training do not work for a dataset with a complete different set of object classes. Our attempts to train Mask-RCNN on the photos of human decomposition had extremely poor performance (even with transfer learning) due to incomplete set of annotations, approximate bounding boxes, and relatively few labeled instances per class.

Other approaches to reduce the annotation effort involve using weakly annotated data, image-level or object- level annotations, for object detection [13, 54, 39] and semantic segmentation [70, 106, 68, 88]. Although these approaches have been successful to some extent, there is still a large performance gap between the models trained on full segmentation masks and those trained on image-level or object-level labels.

The main goal of this work is to simplify and speed-up the annotation process of forensic imagery by developing a machine-assisted semantic segmentation system called Proposed Annotations (PA) that recommends potential annotations to annotators to simply accept or decline.

Semantic segmentation needs a large training set. Our technique [101][2] relies on the fact that human decomposition images are dominated by texture-like patterns (Figure 3) repeated throughout a class. Our method, therefore, can work with a simple classifier and small training data. It utilizes the classifier in combination with a region selection technique to produce potential annotations and presents them to expert annotators.

In addition, our approach can be used to estimate probabilities of a specific forensic class being present in un-annotated images. While this is possible with other semantic segmentation methods, it is of particular use in forensic data, where a major problem faces the annotator: how to design sampling strategies to select images for manual annotation from the collection of one million images.

Therefore, our contribution in this work is twofold. First, we present a novel semantic segmentation technique using a classifier and a region selector for forensic data, leveraging their pattern-like nature. Second, we use this method to propose not only new regions of interest for annotation, but also new images that are likely to contain classes of interest.

B.2 Proposed Annotations (PA)

PA is comprised of a classifier and a region selection method. The classifier is trained on images that contain a single class. It is then used along with a region selection method to detect regions of new images. The classified regions are then merged into larger segments resulting in semantic segmentation. An overview of this process is shown in Figure 4.

This process has three main steps: data preparation (Section B.2), classification (Section B.2) and semantic segmentation (Section B.2).

Human Decomposition Dataset

Our image collection includes photos that depict decomposing corpses donated to the Forensic Anthropology Center at the University of Tennessee. The photos are taken periodically from various angles to show the different stages of body decomposition. The collection spans from 2011 to 2016, and has over one million images.

[2] Mousavi, Sara, et al. "Machine-assisted annotation of forensic imagery." 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019.

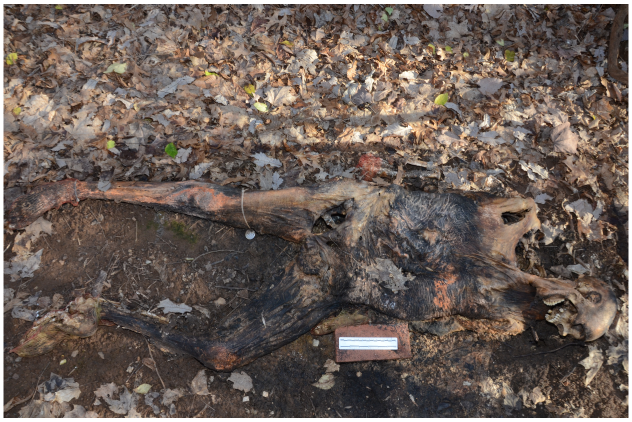


Figure 3: A sample image from the human decomposition dataset. The image highlights the texture-like nature of the data. The image resolutions vary from 2400 × 1600 up to 4900 × 3200.

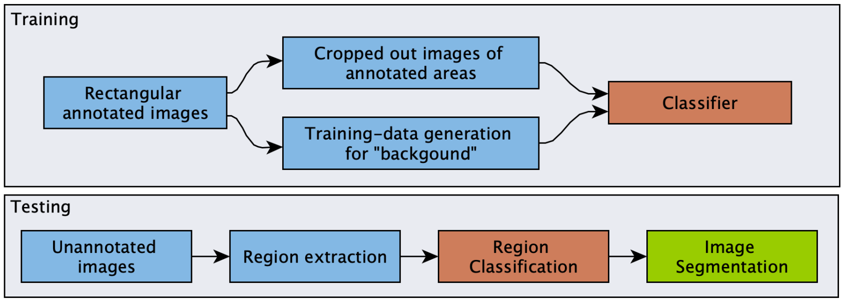

Training
Rectangular annotated images
Cropped out images of annotated areas
Training-data generation for "backgound"
Classifier
Testing
Unannotated images
Region extraction
Region Classification
Image Segmentation

Figure 4: An overview of the structure of PA. Blue, orange and green boxes represent data preparation, classification, and segmentation stages respectively.

The annotation for a small subset of this dataset has been done manually by four forensic experts resulting in 2865 annotated images. However, as previously mentioned, these images are not fully annotated.

A sample image from the human decomposition dataset is shown in Figure 3. The cadaver is mostly camouflaged in the background patterns.

The forensic classes used in this work along with the number of annotated instances for each, are shown in Table 1.

Manual annotation

To enable manual annotation of the small subset, we built an online platform that allows browsing, querying, and annotating ITS-HD. The annotator starts by first selecting a rectangular bounding box around the region of interest and then enters the appropriate class name in an input dialog. The bounding boxes' coordinates along with the class names are stored in a database.

Data Preparation

Preparing training data is a crucial step for making a highly accurate classifier. Due to the similarity of some forensic classes to the background, both in terms of color and texture, we added an additional class to the actual forensic classes for "background". We then cropped areas designated as the forensic classes from the annotated images and used the class name to label each cropped section. Therefore, each annotation became a new training image by itself. For the images cropped for "background", in order to create a diverse range of training data, we used a sliding window to extract smaller images from each training image. We re-sized all images to 224*224 and, as is commonly done, we also generated additional training data from the existing annotations using data augmentation.

Classification

We used a CNN with a multinomial logistic regression classifier to train a model for classifying regions of the un-annotated images. The preponderance of texture-rich classes did not call

Table 1: An overview of the forensic classes of the human decomposition dataset. The number of annotated instances is shown.

Class name	#samples	Class name	#samples
maggots	1375	eggs	533
scale	716	mold	339
purge	709	marbling	241
mummification	557	plastic	107

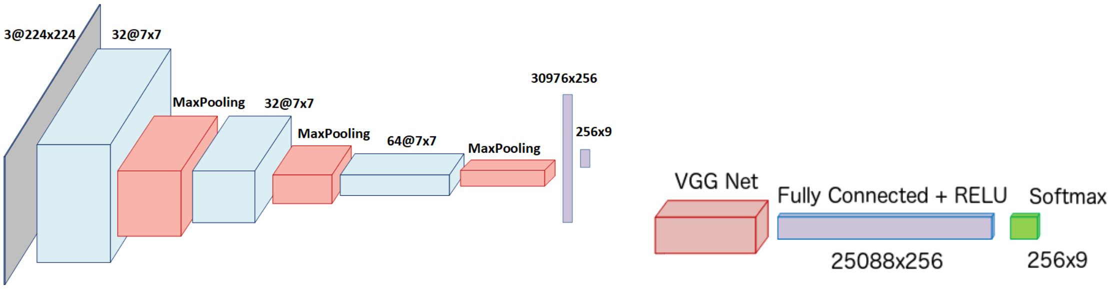


Figure 5: The architecture of Model1 (left), and the VGG-based Model2 (right) is shown.

for very deep neural networks. We started with Model1 that uses a simple neural network shown in Figure 5: left. The CNN network in this model has three convolutional and two fully connected layers. We used normalization after each layer and also a drop-out of 0.5 before the last layer. In addition to Model1, we also experimented with Model2 , a standard VGG [123] with two fully connected layers added on top (Figure 5). Images generated from section B.2 were used to train and validate these two models. We trained Model1 from scratch. However for Model2, we tested both pre-trained weights obtained from ImageNet as well as random weights.

Semantic Segmentation

Locating the forensic objects within images is done using the classifier described in section B.2. Algorithm 2 shows how semantic segmentation is done in PA. Regions of un-annotated images are fed into the classifier model to be classified. The regions are generated by sliding a window of size 224*224 with a stride of 200. Since the training data is not fully annotated, many regions within an image may contain classes that the classifier has not been trained on. To reduce the number of such false positives, we use a threshold of 0.85 to accept a classification done on a region, otherwise it will be ignored.

The contiguous classified regions of the images need to be organized so that neighbor regions belonging to the same class are proposed as a single composite segment. To do so, we group the classified regions by first finding overlaps. Then, we create an adjacency matrix of size $n \times n$ where n is the number of regions for the class. A cell (i, j) (for two regions i and j) is set to 1 if the two regions overlap. We then create a graph from the adjacency matrix and find the connected components of the graph for each class using the iGraph library [32]. Next, we find the convex hull for each connected component. The resulting hulls are presented to the annotator as proposed annotations. The confidence of a recommended annotation is calculated based on the average confidence of the individual regions within that component.

```
Result: Semantic segmentation using PA
for every region in image do
    Classify(region)
    Store region's coordinate, class_id and confidence
end
for every c in classes do
    Find all regions classified in class c
    Create an adjacency matrix of regions
    Create connected-components to group neighboring regions
    Draw the convex hull of each group
    Calculate score for each colored area
end
Present the segmentation as proposals to the annotator
```

Algorithm 2: Semantic segmentation using PA

B.3 Results and Discussion

To evaluate PA, we measure the accuracy in comparison to the manual annotation done by a forensic expert. The results include the performance of Model1 and Model2. We also tested the effects of including the background as a separate class in both models and also the effect of transfer learning on Model2. Section B.3 describes tuning parameters for both models and evaluation setup. Section B.3 discusses our findings.

Evaluation Setup

PA is implemented using Keras, TensorFlow and Python. We used MongoDB as our database. For both CNN networks we used the *SGD* optimizer with a learning rate of 0.001.

Over two hundred distinct classes of samples were present in the dataset. To select a more manageable number of classes for the experiments, we first excluded classes with fewer than 100 ground truth instances and asked forensic experts to select the most important classes for the forensic community. We used one third of images per class for validation and the remaining images for training.

To evaluate the performance of our PA, we randomly selected 46 images and asked a forensic expert to provide us with the ground truth annotation masks only for the forensic classes used in this work. These images were annotated carefully and completely with polygonal selections, taking about 3 hours to complete. We evaluated the performance of our proposed annotations against these round truths.

Discussion

Table 2 shows the performance of PA. We calculated mean average precision (mAP) for the classification done by both Model1 and Model2 over all classes. We also calculated mean average recall and precision over all classes (mAR, mAP) for our semantic segmentation against the ground truth. These values are used as mAP and mAR in Table 2.

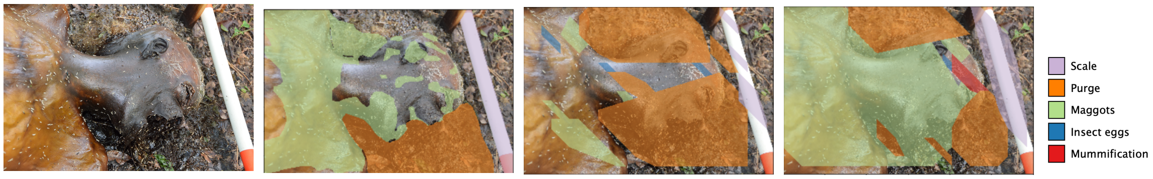
The mean average precision is calculated as the ratio of correct predicted pixels over the total predicted pixels for each class. This value is then averaged for each class over all 46 images. We used a similar method for mAR, however we used the the ratio of correct predicted pixels to the total ground truth pixels for each class.

Table 2 shows that transfer learning improves the performance of Model2. Comparing Model2 with Model2-tl, we can see that transfer learning has improved both mAP of the classifier model and mAR of the semantic segmentation.

Comparing Model2 with Model1 in Table 2, we believe that we might get even better results using Model1 if we first train it on another dataset such as ImageNet, considering the fact that Model1 is a very simple model and its training takes less time compared to Model2.

A trade off between using a model with high recall or high precision can also be observed from the table. For the purpose of suggesting classes to a human annotator, it is more important to detect a forensic class if it exists, as opposed to exactly pinpointing the location of the class within the image. Thus, we want to have a model with higher recall and a reasonable mAP. Our results also show that including the background as a class improves mAP for segmentation.

Figure 6 shows a segmentation using Model1 without transfer learning and Model2 with transfer learning, and compares it to the ground truth. Both models were trained on 8 forensic classes plus the background class. We can see that a better segmentation is obtained when transfer learning is employed.



a: Original image b: Ground-truth c: Model1's result d: Model2's result

Figure 6: Detected forensic classes using Model1-bg and Model2-bg-tl are shown in (c) and (d) respectively. Sub-figures (a), and (b) show the original image and the ground truth respectively. Concave annotations are a result of overlaps between two convex hulls where one overlays part of the other.

Table 2: Performance of classifier models and semantic segmentation in PA. bg and tl stand for background and transfer learning.

	Semantic Segmentation		Classification
Method	mAP	mAR	mAP
Model2-bg-tl	0.26	0.45	0.95
Model2-tl	0.15	0.59	0.92
Model2	0.30	0.28	0.79
Model1	0.16	0.32	0.84
Model1-bg	0.17	0.23	0.88

B.4 Conclusion

In this work, we discuss an annotation-assistance system that proposes annotations within an image as well as images likely to contain a desired class to forensic experts. At the core of our system we introduce a semantic segmentation method composed of a classifier in conjunction with a sliding-window-based region selection method. We also evaluate its applicability in the context of imagery documenting human decomposition where classes are primarily determined by patterns. We demonstrate the feasibility of semantic segmentation in this domain using a relatively small set of training samples. As is expected with small training samples, transfer learning has been effective. Inclusion of the background as a class also brought improvements, possibly because background is at times difficult to distinguish from focal classes.

In the future, we would like to evaluate if our method would work with other types of texture-like data. In addition, we plan to utilizing body pose detection methods to improve the ability to exclude background and increase the accuracy of our system for forensic class segmentation.

Vita

Sara Mousavi received a BS in Computer Engineering from Razi University, Iran in 2012. She started working under Dr. Chao Tian at The University of Tennessee, Knoxville (UTK) in the area of Information Theory in 2016, and received her MS degree in 2017. She then started working under Dr. Audris Mockus in the area of applied machine learning for curating large datasets in 2018. During her studies at UTK, she completed a Summer internship at Verizon where she led the team of interns and worked on developing and delivering an end-to-end platform for data storage, retrieval and modification between IoT devices and AWS. In December 2021, Sara graduated with a Doctor of Philosophy degree in Computer Science with a focus on applied machine learning. Sara's research interests include data analysis and facilitating the process of gaining insight and knowledge from data through computer vision, machine learning, and data science techniques.