

CMed: Crowd Analytics for Medical Imaging Data

Ji Hwan Park¹, Saad Nadeem, Saeed Boorboor², Joseph Marino², and Arie Kaufman², *Fellow, IEEE*

Abstract—We present a visual analytics framework, CMed, for exploring medical image data annotations acquired from crowdsourcing. CMed can be used to visualize, classify, and filter crowdsourced clinical data based on a number of different metrics such as detection rate, logged events, and clustering of the annotations. CMed provides several interactive linked visualization components to analyze the crowd annotation results for a particular video and the associated workers. Additionally, all results of an individual worker can be inspected using multiple linked views in our CMed framework. We allow a crowdsourcing application analyst to observe patterns and gather insights into the crowdsourced medical data, helping him/her design future crowdsourcing applications for optimal output from the workers. We demonstrate the efficacy of our framework with two medical crowdsourcing studies: polyp detection in virtual colonoscopy videos and lung nodule detection in CT thin-slab maximum intensity projection videos. We also provide experts' feedback to show the effectiveness of our framework. Lastly, we share the lessons we learned from our framework with suggestions for integrating our framework into a clinical workflow.

Index Terms—Crowdsourcing, medical imaging, virtual colonoscopy, lung nodules, visual analytics

1 INTRODUCTION

THE prevalence of non-invasive imaging techniques such as computed tomography (CT) and magnetic resonance imaging (MRI) has significantly increased the amount of patient data available to radiologists for interpretation. Double reading (two or more radiologists interpreting the same examination), computer-aided detection (CAD), and visualization techniques have been proposed to facilitate interpretation and expedite the decision-making process for the radiologists. Due to a lack of resources, double reading is not normally used in clinical practice. CAD algorithms can still miss life-threatening cancerous lesions, and it is mandatory to keep physicians in the loop while looking for anomalies in patient scans.

Crowdsourcing seeks to engage the general masses in innovative ways and solicit their inputs in solving diverse problems. Previous studies have shown that most non-expert crowd users (workers) are forthright in their intentions [1]. Crowdsourcing has shown promise in medical annotation tasks [2]. These attempts open up avenues to incorporate crowdsourcing as an additional tool along with CAD, double reading, etc. in the clinical workflow to assist radiologists in the critical task of abnormality screening.

There is little prior work on studying the performance of crowd workers in medical annotation tasks in detail. Some preliminary insights suggest that there is a task dependent bias in crowdsourcing [3], and thus crowd workers might be good at detecting anomalies in some videos but not all. Additionally, since not every worker produces good quality

annotations (worker's bias), one needs to filter out spammers to obtain good results [4]. Moreover, previous work has shown that workers' behavior patterns can have an effect on accuracy [5]. In order to facilitate a crowdsourcing analyst/developer in designing a better crowdsourcing application based on previous crowdsourced annotation data, the worker and task-dependent biases and worker's behavior patterns need to be analyzed. None of the previous tools for visualizing crowdsourcing data [5], [6], [7] meet all these requirements and hence, we present CMed for observing patterns and gathering insights into crowdsourced medical data, in detail not previously possible. We also provide lessons we learned from designing our framework and exploring the output of CMed. Based on the insights from CMed, future crowdsourcing studies can be designed for optimal output from the workers. We also suggest how the output of our framework integrates into a clinical workflow.

CMed is a visual analytics framework used to visualize, classify, and filter crowdsourced clinical data (Fig. 1). More specifically, we use the ground truth from medical experts, crowd annotations, and the logged events of the crowd workers as our source of input. We compute the accuracy of the crowd annotations and cluster these to display (in a compact view) how good the workers are. We also extract each worker's logged events and cluster these to observe the effect of workers' behavior patterns on the quality of their video annotations. We offer several interactive linked visualization components for presenting different aspects of crowd annotations. The target users of CMed are crowdsourcing analysts/developers who are responsible for designing crowdsourcing applications and managing workers for medical data.

The main contributions of this paper are summarized as follows:

- We provide an interactive visual analytics framework to visualize, classify, and filter crowdsourced clinical data, helping developers understand the crowd, improve their current crowdsourcing framework, and design future crowdsourcing framework.
- We offer a set of visualization techniques to support exploring different aspects of crowd annotations.

• J.H. Park is with Brookhaven National Laboratory, Upton, NY 11973-5000 USA. E-mail: parkj@bnl.gov.

• S. Nadeem is with Memorial Sloan Kettering Cancer Center, New York, NY 10065 USA. E-mail: nadeems@mskcc.org.

• S. Boorboor, J. Marino, and A. Kaufman are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-2424 USA. E-mail: {sboorboor, jmarino, ari}@cs.stonybrook.edu.

Manuscript received 27 Dec. 2018; revised 24 Oct. 2019; accepted 6 Nov. 2019. Date of publication 20 Nov. 2019; date of current version 29 Apr. 2021. (Corresponding author: Ji Hwan Park.)

Recommended for acceptance by J. van Wijk.

Digital Object Identifier no. 10.1109/TVCG.2019.2953026

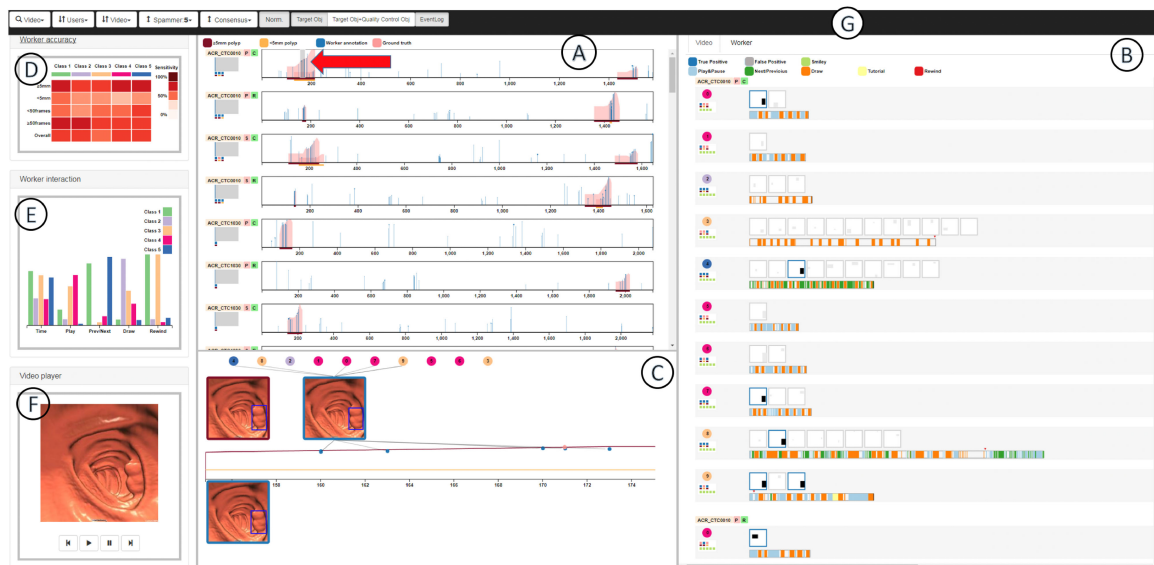


Fig. 1. The CMed system: (A) Timeline View displays a summary of annotations for each video. (B) Worker View shows workers' annotation and the corresponding event patterns. (C) Frame View presents details of selected frames. (D) Matrix View shows the correlation between users' event patterns and their accuracy. (E) Class View displays characteristics of worker classes based on event patterns. (F) Video View shows a selected video, and (G) Control Panel for selecting and reordering data. Views are linked, e.g., selected frames for the top video in the Timeline View (A), highlighted with a gray bounding box (pointed to by a red arrow), are shown in the Frame View (C), and the same selected frames are also highlighted in the Worker View (B).

- We characterize a set of application-specific goals and design requirements derived through discussion with medical experts and previous crowdsourcing studies.
- We demonstrate the efficacy and effectiveness of our framework with two case studies: (1) polyp detection in virtual colonoscopy (VC) videos, and (2) lung nodule detection in CT thin-slab maximum intensity projection (MIP) videos.
- We provide guidelines and lessons we learned from our framework, as well as suggest how to integrate our framework into a clinical workflow.

2 RELATED WORK

Crowdsourcing approaches are popular in various domains such as image classification and labeling and video annotations. Even though most workers are honest and diligent, some workers are dishonest and/or less skilled than others [8]. To detect these workers and improve workers' output, several approaches have been proposed. The most popular approach is adding verifiable questions in a task [8]. If workers answer those questions, they are considered honest workers. However, it is difficult to design good verifiable questions for complex tasks. If verifiable questions are not well designed, some workers can focus on the verifiable questions and cheat on the actual questions/tasks [9]. Another approach is to aggregate workers' answers (crowd consensus) such as majority voting [10] and Generative model of Labels, Abilities, and Difficulties (GLAD) [11]. In some cases, including our input data, workers can annotate the same object (a polyp/nodule in our case) on different frames, which can be relatively far apart. Thus, we are not able to aggregate answers because we cannot ascertain whether two annotations at different frames are for the same object.

These approaches have been utilized in medical crowdsourcing applications. One crowdsourcing study added 5 verifiable questions at the beginning, where workers identified

whether an area in an image was air, tissue, or fluid [12]. Only workers who correctly answer at least 4 out of 5 questions can participate in the study. Majority voting has been used to correct for low-quality work [13], where a task was to determine whether there was a polyp or not in each video segment. In these two studies, existing quality control methods worked well, but they are not appropriate for complicated cases. The reason is that a worker can focus on only the verifiable question at the beginning of a task, or only a few skilled workers can detect an object (missed by the majority of workers) because a target object is difficult to be detected by novice workers. Thus, a more sophisticated quality control approach is needed.

In medical applications, obtaining ground truth annotations is difficult, so it is hard to apply deep learning approaches. In recent years, weakly-supervised video annotations [14], [15] have shown promising results, where only point annotations are required. However, as shown in [15], the performance of the algorithm can be further improved if bounding box labels are provided. Thus, crowd annotations may be deployed for improving such approaches.

In our CMed framework, we use crowdsourced clinical data, where the crowd workers view videos created from medical image data. There are several approaches for visualizing crowdsourced data and video-based data. Viz-A-Viz [16] uses basic computer vision techniques to classify datasets of human-activity from a large volume of surveillance videos and couples the aggregated sequences with information visualization components to allow for high-level human analysis. Considering the major events in a video, 3D visualization techniques were used to develop a framework for hierarchical event representation and an importance-based event selection algorithm to create a video storyboard [17]. In addition to visualizing the summary of the video content, an analytics system was introduced for interactive exploration of individual actions as well as the trajectories of moving objects, as a space-time cube, in surveillance videos [18]. An important visualization component for highlighting areas of interest in video analytics is annotations. Typically, annotating each area

of interest is a time-consuming step of the analysis process. This was addressed by proposing a visual analytics approach through an image-based, automatic clustering method [19]. In particular, they allow direct interpretation of the labeled data by coupling annotation and analysis components using multiple linked views. However, these works focus on analyzing a single video or videos for a single scene and thus cannot be used for our target crowdsourcing applications, where workers view different videos and annotations on videos and workers' event logs should be analyzed.

Visual analytical frameworks for crowdsourced medical applications is not a widely explored theme. To present a framework for clustering and interpreting results from the crowd, Willett et al. [20] proposed a system for analysts to interactively examine the workers' insight by clustering worker explanations and capturing workers' browsing behavior via an embedded web browser. Similarly, CrowdScape [5] and Mimic [21] evaluate the quality of the workers' answers based on their behavior and present a visualization tool to interactively explore these features, enabling users to classify workers. The latter help interaction designers understand the relationship between workers output and their behavior by focusing on micro interactions. There is a visual analytics platform to visualize crowdsourced survey data with multiple choices by using glyphs and parallel coordinate plots [6]. Recently, C²A [7] was developed to visualize crowdsourced medical data, where a worker viewed twenty video segments and answered whether a polyp is present or not in each segment. The main difference between C²A and CMed is that the input for C²A is a simple binary label for each video segment, while CMed has crowd annotations and workers' logged events as the input. Moreover, the goal of C²A is building crowd consensus, while CMed focuses on how the crowd annotates target objects and on improving a crowdsourcing framework.

In crowdsourcing, there is a task dependent bias. For example, workers can be good at labeling some images while they can fail to label other images even if all images belong to the same image category, such as galaxy images [3]. Additionally, workers have different backgrounds, skill levels, and motivations, and these biases result in different quality for the outputs of their work [4]. All studies mentioned above focused on workers' behavior patterns or either one of these biases. However, in our framework, a crowdsourcing application analyst can explore all these elements (user behavior pattern analysis, task dependent bias, and workers' annotation history), which were offered separately in previous work. Moreover, to the best of our knowledge, there is no visual analytics framework to explore crowdsourced medical data, where the crowd annotates target objects such as polyps/nodules in videos. Our CMed platform enables a crowdsourcing application analyst to observe patterns and gather insights into crowd annotations in the crowdsourced medical data, helping the analyst to design better crowdsourcing applications.

3 BACKGROUND AND INPUT ANNOTATIONS

Our target applications are virtual colonoscopy and lung nodule detection, each of which requires data generation, inputs for the crowdsourcing platform, and a process to obtain crowd annotations.

3.1 Background

Virtual Colonoscopy Videos. Virtual colonoscopy (VC) is a non-invasive procedure for detecting polyps, the precursors of colon cancer, in CT data. A radiologist flies through a 3D colon

(reconstructed from abdominal CT data) and inspects the colon wall for polyps, characterized by bumps on the wall. On average, a complete inspection of the colon in two different patient orientations (e.g., supine and prone) from rectum to cecum and back takes approximately 15-30 minutes to perform. In our previous studies, we have shown that this tedious bump detection task can easily be relegated to non-expert workers [13]. Note that optical colonoscopy is still the gold standard for colorectal cancer screening in many countries and VC costs are usually not covered by health insurance, though this is changing due to the higher patient compliance rate with VC and the many advantages of VC.

Lung Nodule Detection. Radiologists interpret 2D chest CT scans to look for lung nodules, the precursors of lung cancer, characterized by isolated "spots" not connected to the prevalent vascular structures. As shown in our lung nodule detection study [22], maximum intensity projection (MIP) videos of these 2D chest scans can help clearly delineate these "spots" for non-expert workers.

Crowdsourcing. In crowdsourcing applications, there are three types of workers [9]: good, bad, and ugly workers. Both good and bad workers complete a task honestly. A good worker understands the goal of a task well and has a good skill to complete a task. However, a bad worker has a poor skill or misunderstands the goal of a task, so his/her output is not as good. An ugly worker cheats on a task, e.g., randomly answering a task. To filter out ugly workers, one of the common techniques is adding verifiable questions in a task (gold standard questions) [8]. If a worker answers the questions correctly, we assume that he/she is not an ugly worker. We added quality control objects to our input data and asked workers to detect those objects as gold standard questions.

3.2 Medical Data Annotations

In this paper, we use crowdsourced annotation data from our two previous studies [22], [23]. In the first work, we used VC videos, and the second work used lung CT videos. The goal of each study was to detect target objects (polyps for VC videos and lung nodules for lung CT videos).

VC Video Generation. VC fly-through videos were generated using the commercially available FDA-approved Viatronix V3D-Colon VC system [24]. Four centerline fly-through videos were automatically generated for each patient VC dataset (from rectum to cecum and from cecum to rectum in both supine and prone orientations). The videos were captured at 15 frames per second (fps) with a resolution of 256×256 pixels and a 90 degree field-of-view. Anonymized datasets from 14 patients were used, generating a total of 56 VC videos. The datasets contained both large (> 5 mm) and small (< 5 mm) sized polyps. There were a total of 33 polyps, which included 10 polyps of less than 5 mm in diameter. We generated the ground truth annotations by marking polyps in the videos based on the expert radiologists' VC reports.

Lung Video Generation. Lung CT videos were generated by rendering videos of overlapping thin-slab MIPs (TS-MIPs) of CT slices through each half of the patient's left and right lungs. MIP is a projection of the voxel with the maximum intensity value along rays traced from the viewpoint to the image plane [25]. For this paper, we used 15 videos from anonymized chest CT patient scans from the publicly-available LIDC database [26], containing 45 nodules. Of the 45 nodules, 19 were ≤ 4 mm, 8 were > 4 and ≤ 6 mm, 3 were > 6 and ≤ 8 mm, 5 were > 8 and ≤ 10 mm, and 10 were ≥ 10 mm in diameter. We generated the ground truth

by marking lung nodules based on 5 expert radiologists' manual annotations.

Crowd Annotations for Both Datasets. To obtain crowd annotations for the VC and lung CT videos we used a web interface, VATIC [9], for both datasets on the Amazon Mechanical Turk platform. When workers first select this task, for each video, a video is downloaded in the background. In the meantime, they are provided with brief instructions including the main objective of the study, how to use the system, and some example target objects (polyps for VC videos and lung nodules for lung videos). After the video download is finished, and the workers have read the instructions, the workers can play, pause, and rewind the video at will. Additionally, they can step to the previous or next frame. All of these interactions are done by clicking corresponding buttons. We added quality control objects (5 smileys for VC videos, 1 gorilla for lung videos) in each video to help detect spammers. When the workers find a target/quality control object, they first label it as a target or quality control object and then annotate the object by drawing a rectangle around it. We logged all interaction events and stored all annotation information. A worker was not allowed to complete the same task multiple times, but was allowed to complete multiple different tasks/videos.

4 CMED FRAMEWORK

In this section, we describe our design requirements concerning CMED and an overview of our framework.

4.1 Design Requirements

Based on our preliminary work and previous crowdsourcing studies [22], [23], we characterized several design requirements to satisfy the following goals: G1) improve training cases based on exploring false positives of workers, G2) explore missed polyps and see why workers missed them, G3) explore the effect of missing quality control objects on sensitivity and type of worker, G4) improve the current interface based on analyzing workers' event logs.

R.1 Compare the ground truth and workers' annotations (G1,G2). Even though a video contains target objects, workers sometimes cannot find them because they are too small, only appear in a few frames, or appear similar to background structures. Additionally, workers might also annotate objects which are not target objects, but look like target objects. Thus, an analyst needs to understand the characteristics of crowd annotations by comparing details of crowd annotations to ground truth annotations. We try to answer the following questions: *How many target objects (polyp/nodule) per video did workers find/miss? Did workers mark an object that was not a target object?*

R.2 Reveal the details of workers' annotations and event logs for each video (G1,G2). The main difference between a bad worker and an ugly worker is that an ugly worker marked a region where there is no object that looks like a target object. An analyst can see whether a worker is a bad or ugly worker by exploring individual annotations from him/her. Additionally, a worker's event logs can show the type of worker. By visualizing this information, we could answer the following questions: *How many target objects did a worker miss? What kinds of objects did a worker annotate? How did a worker annotate/explore a video? Is he/she an ugly worker? What did an annotated object look like in a video?*

R.3 Reveal the overall quality of each worker's output and his/her event patterns for multiple tasks (G3). In order to exclude ugly workers, workers in our input data were

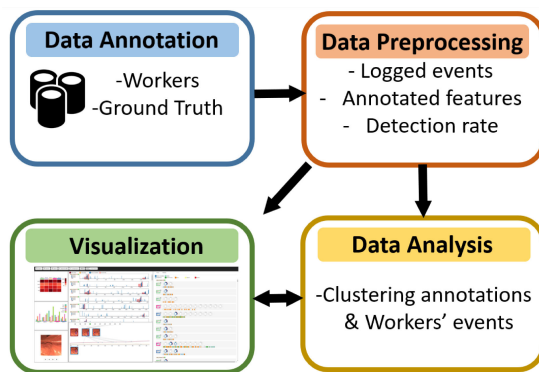


Fig. 2. Overview of the CMED framework pipeline. We first collect both crowdsourced and ground truth annotations. We then extract the workers' logged events and annotated features and compute the detection rate of the target objects. Next, we incorporate several approaches to cluster the workers' annotations and their events. Lastly, we visualize all the data for interactive exploration.

asked to detect quality control objects. However, some workers may be good at detecting target objects, but miss some quality objects. Workers can annotate multiple tasks/videos and they can be good workers even if they missed several target objects and/or if they are better than other workers. A worker can be a good worker if the sensitivity of the worker is higher than the overall sensitivity of all workers for multiple tasks. Additionally, there might be a learning effect on their results throughout tasks. The workers also can change their interaction behavior to annotate target objects depending on a video. Event patterns refer to how a user annotates an object (e.g., only using play and stop buttons, or never rewinding a video). By visualizing this information, we could answer the following questions: *Does a worker have similar event patterns throughout multiple tasks? Is the accuracy of a worker's annotations changed throughout tasks? Is the sensitivity of the worker higher than the overall sensitivity of all workers for multiple tasks? Does a worker need to be excluded if he/she was an ugly worker based on the number of detected quality control objects in a task?*

R.4 Discover the correlation between workers' event patterns and the sensitivity of corresponding annotations for all datasets (G4). There might be a good strategy to annotate target objects in videos from medical image data. If there is such a strategy, it might improve the sensitivity of workers by providing workers with only these interactions (e.g., if a worker rewinds a video, he/she may perform better than a worker who just plays a video and annotates target objects). For example: *Do workers' event patterns affect the sensitivity of their answers? Which event pattern/class is best/worst for each type of target objects?*

4.2 CMED Overview

CMed is a web-based application developed under the framework of Express.js. Annotation data as our source of input is stored in MySQL. The data preprocessing module was developed in JavaScript, and the data analysis module is developed in JavaScript and Python with OpenCV. The visualization module is implemented in D3.js. Our CMED framework consists of three major components (Fig. 2): data preprocessing, data analysis, and data visualization.

4.2.1 Data Preprocessing

In order to analyze and visualize the input annotation data, we need to extract and classify data as follows:

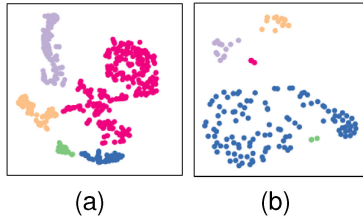


Fig. 3. Examples of clustering workers' logged events using t-SNE and single-linkage clustering: (a) VC datasets, and (b) lung CT datasets. Different colors indicate different clusters.

Logged Events. In the crowdsourced annotation data that is the source of our input, we logged the types of events that would occur as the workers performed the task: playing and pausing a video (*Play*), stepping to a next/previous frame (*Next/Previous*), rewinding a video (*Rewind*), and events related to annotations (*Draw*) such as drawing, resizing, and dragging and dropping a rectangle with timestamps. We first calculate the time for each task per worker (*Time*) by using the difference between the timestamp on the last event and the timestamp on the first event performed by the worker after the instructions have been read and the video has finished loading. We then align the workers' annotation data with the logged events based on the timestamps of events.

Detection Rate of Target Objects. We find all ground truth annotations present in the same frame as the workers' annotations and then used the Dice Similarity Coefficient (DSC) [27] to determine whether or not a worker annotation W corresponds to an actual ground truth annotation T for both target objects and quality control objects. The DSC is calculated as

$$DSC_{WT} = \frac{2|W \cap T|}{|W| + |T|}. \quad (1)$$

A worker annotation is considered a match to the ground truth if $DSC_{WT} > 0.5$. Since the ultimate goal of crowdsourced medical data is to find a suspicious area and show it to a radiologist, even a partial match can be a good indicator to show a suspicious area.

Workers' Annotation Features. Based on the accuracy of the workers' annotations, we store matching ground truth target objects and the centroid of the annotated regions. We also calculate the number of missed quality control objects for each worker.

Ground Truth Annotation Features. Unlike the workers' annotations, where a polyp/nodule was only annotated in a single frame, the ground truth annotations per target object were present in multiple frames. To analyze the characteristics of these annotations in our data analysis component, we first compute the number of annotated frames. We then compute the average area size A_i and ratio R_i of annotations J per target object i as following:

$$A_i = \sum_{j \in J} (w_{ij} * h_{ij}) / N_J, \quad R_i = \sum_{j \in J} (w_{ij} / h_{ij}) / N_J, \quad (2)$$

where w_{ij}, h_{ij} are the width and the height of an annotation j , respectively, and N_J is the number of annotations J .

4.2.2 Data Analysis

We analyze preprocessed data to perform several tasks (R.2-4). Additionally, we detect an ugly worker by calculating that a worker missed a certain number of quality control objects, where this number is selected interactively by an analyst.

Clustering Workers' Annotations. In our crowdsourced annotation data, there are many annotations for the same object. Thus, we need to cluster these annotations to aid an analyst in analyzing the annotations (R.2). In our data, each annotation can be matched to annotations in the same frame or the closest frame with high probability. Thus, we first search annotations in the same frame and compute whether they are matched or not. We then compared annotations in the current frame to annotations in the closest frame. To determine whether they are matched or not, we used two approaches. The first method is extracting scale-invariant feature transform (SIFT) features [28] within an annotated area and then using brute-force matching. SIFT features are scale, orientation/rotation, illumination, and (partially) viewpoint invariant. SIFT consists of four steps: 1) feature point detection, 2) feature point localization, 3) orientation assignment, and 4) feature descriptor generation. We chose the SIFT method because it is a current state-of-art method and works well for our input data [29]. We experimented with various annotation sizes. If an annotation is too small (the area of an annotation ≤ 900 pixels in our target datasets), matching using SIFT features sometimes fails to find a match. Therefore, we use another approach using the centroids and frame distances of two annotations i and j as follows:

$$Match_{i,j} = \|C_i - C_j\| < \alpha \quad \text{and} \quad |F_i - F_j| < \beta, \quad (3)$$

where C_i, C_j are the centroids and F_i, F_j are the frame numbers of i and j , respectively, and α and β are user-defined constants. We empirically set 10 for α and 5 for β in our case studies. We use OpenCV for this calculation.

Clustering Workers' Logged Events. To compare workers' event patterns and discover the effects of workers' event patterns (R.2-4), we need to cluster the logged events from the annotation tasks. We first create a vector containing the events (*Play, Next/Previous, Draw, Time, Rewind*) for each worker and then run a dimension reduction method, t-SNE [30] with the euclidean distance similarity metric to preserve the local and global structure of the data. To cluster workers' logged events, we provide an analyst with t-SNE to find the number of distinct event classes (5 clusters in our case). After setting the number of the classes, we use single-linkage clustering [31] to obtain event classes automatically, which is a hierarchical agglomerative clustering method. For our data, our clustering method successfully detects these five classes (Fig. 3). We tried k -means and hierarchical clustering, but the latter method showed clusters more clearly in our target dataset. Other clustering methods [32] may perform better for other datasets.

5 CMED DESIGN

Based on our design rationales (R.1-4), we designed our framework to contain several linked views. The *Timeline View* provides an overview of workers' and ground truth annotations. To see details of these annotations, we offer the *Frame View*, which shows details of selected frames from the *Timeline View*. In the *Worker View*, we provide two types of information: each workers' annotations and the event patterns for each video, along with a summary of this information for all completed tasks by each worker. The *Class View* and the *Matrix View* aid a crowdsourcing application analyst in understanding the correlation between workers' event classes and their corresponding accuracy. The *Video View* allows a crowdsourcing application analyst to investigate the context of a target object in a selected video. The main views in the

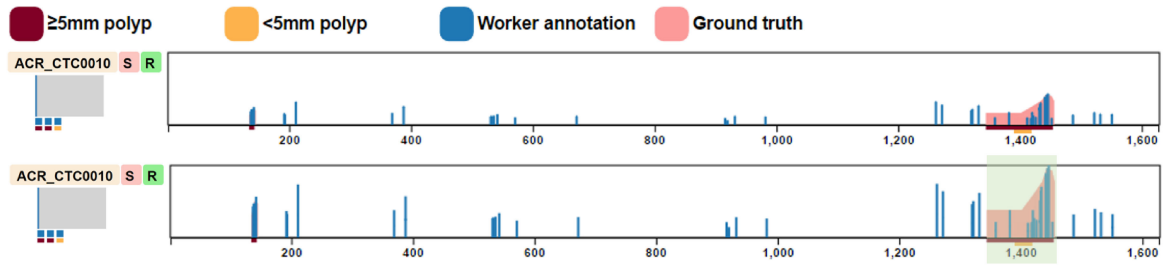


Fig. 4. An example of two modes in our Timeline View: Area mode (top) and normalized mode (bottom). The two modes show the same annotations for each video, but have different scales for the bar heights. In both modes, the x -axis is the video timeline, indicating the video frame number. Workers' annotations (■) and ground truth annotations (■) are shown as vertical bars. The details of the highlighted (■) area are described in Fig. 5.

CMed framework are the Timeline View, Frame View, and Worker View. The other views (The Matrix View, Class View, and Video View) can be hidden if they are not necessary. We used the ColorBrewer color scheme [33] for the visual elements in each view.

5.1 Timeline View

In the Timeline View (Fig. 4), we reveal the overall annotation information for each video ($R.1$). The Timeline View can order videos by the name of a video, by the sensitivity of identified target objects, and by the number of target objects. The Timeline View consists of two parts: a summary of annotations on the left side and annotations per frame on the right side (Fig. 5).

The left side (a blue box in Fig. 5) displays the ratio of annotated frames to the total number of frames in each video by the width of a bar (Fig. 5a). At the bottom of the bar, we display the number of target objects from the ground truth as the number of squares, and the color of each square indicates whether a target object is identified by a certain number of workers (blue: detected, pink: missed); this threshold can be interactively changed by the analyst in our Control Panel. We also place bars at the bottom of the squares, where the color of each bar indicates the size of each target object. We note that the size of the blue bars for the ratio and the colored squares may appear small, but the purpose of these elements is only to show how small the annotations are and whether there are missed target objects.

On the right side (a red box in Fig. 5), the x -axis is the video timeline indicating the video frame number, and the y axis represents the magnitude of the area of an annotation. Annotations are shown as vertical bars with different colors (blue for workers' annotations and pink for ground truth annotations). For each bar, we select an annotation with the largest area among annotations in the same frame. The areas

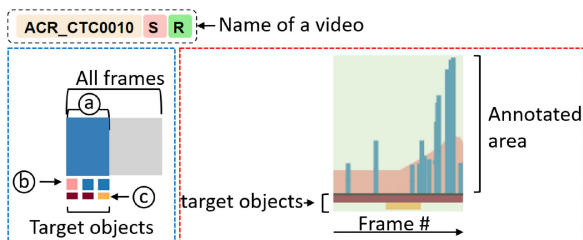


Fig. 5. Timeline view illustration. The left side displays (a) the ratio of annotated frames to total frames in each video, (b) the number of target objects identified (■) and missed (■) by workers, and (c) their size encoded in the bar colors. On the right side, stacked horizontal bars at the bottom of the x -axis show the ground truth target objects, where bar width indicates the number of frames in which the target is visible and color represents target size.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 18, 2024 at 17:53:35 UTC from IEEE Xplore. Restrictions apply.

of some annotations might be too small to be noticeable in the view. Thus, we provide two modes to scale the y -axis: area mode and normalized mode. The area mode shows the actual area of annotations (e.g., how small an annotation is), while the normalized mode shows the relative difference between annotations. In area mode, the height of each bar is scaled based on the maximum possible size of the annotation (i.e., percentage of the full video frame size). In normalized mode, we select the annotation with the largest area in the video and scale the heights of the bars for that video such that the largest area will fill the entirety of the vertical space. Fig. 4 illustrates these two modes.

At the bottom of the x -axis, we stack horizontal bars for each target object from the ground truth, where the width of each bar indicates the number of frames in which the target object is visible in the video, and the color represents the size of a corresponding target object, which was also from the ground truth. The analyst can select specific frames by brushing to see details in the Frame View. The selected frames are also shown in our Video View.

On the right side (a red box in Fig. 5), we visualize only the annotation with the largest area in the same frame; an analyst cannot see a small target object if several targets are overlapped. To help view these overlapped objects separately, selecting a square representing the target object on the left side of the view highlights the area corresponding to that target by graying out the other ground truth annotations.

5.2 Frame View

An analyst can select specific frames in the Timeline View, and the details of the selected frames are then visualized in the Frame View (Fig. 6). This allows analysts to compare

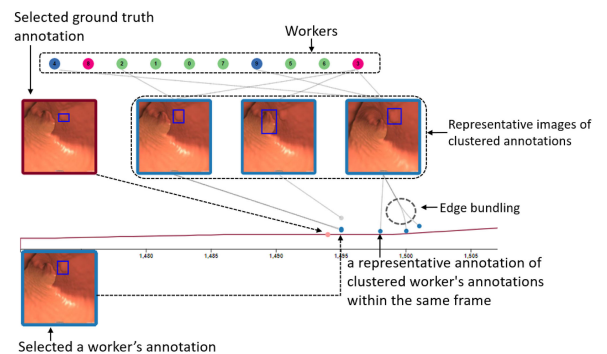


Fig. 6. An example of our frame view. Each line represents a target object; the x -axis indicates the video frame number and the y -axis is the area of the annotation. Representative annotations (dots on the lines) and images are illustrated by clustering workers' annotations within the same frame and across frames, respectively. We can display ground truth and/or a representative annotation of a selected point on a line.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 18, 2024 at 17:53:35 UTC from IEEE Xplore. Restrictions apply.

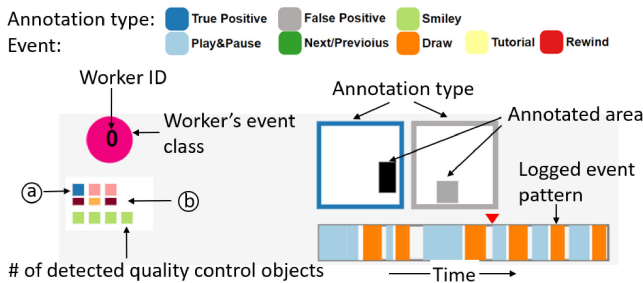


Fig. 7. Illustration of our worker view for a video. The left side displays a summary of a worker's annotations such as (a) the number of identified and missed target objects by each worker and (b) the size of the objects. The right side shows details of the worker's annotations and logged events.

details of the ground truth and the workers' annotations (R.1, R.2). Similar to our Timeline View, there is a chart, where the x -axis indicates the video frame number and the y -axis the area of each ground truth annotation. Each line represents each target object that is present within the selected frames, and the color of each line indicates the size of each target, which is the same as the Timeline View. In order to display annotations from the workers, we first compare all annotations in the same frame by the DSC mentioned earlier. If there are similar annotations, we group them together and store the annotation with the largest area as a representative of the annotations. We then visualize the representatives of the grouped annotations, where each dot's color indicates it is a representative (blue dot) or selected ground truth (pink dot) annotation. The size of each dot (blue dots near the red line as a representative annotation in Fig. 6) represents the number of annotations within the group. We use the same colors we used in the Timeline View. Additionally, if a worker's annotation is not matched to the ground truth, we change the color of the annotation to grey to display that it is a false positive.

We provide the image of a frame with an annotation to show what an actual annotation looks like in a video frame (Fig. 6). We use the clustering method described earlier to cluster worker annotations in neighboring frames since the view can be too cluttered if we display frames for all representative annotations. For each cluster, we display only the annotation with the largest area. We link these clustered annotation images to the corresponding dots for representative annotations. Additionally, the analyst can view a representative annotation by selecting a dot or view a ground truth annotation by clicking any position along the line.

Circles with unique IDs are placed at the top of the view to indicate the workers who completed the selected video. These numbered circles are linked to the clustered annotations to enable the analyst to understand whose annotations are in a cluster. When an analyst selects an annotation circle, the workers who made the annotation are highlighted. We used an edge bundling algorithm [34] to reduce visual clutter between clustered annotations and representative images and between worker ID circles and representative images.

5.3 Worker View

In the Worker View, we visualize two types of information (Figs. 7, 8). The first type is details of the workers' annotations and logged events for each video. In this type, we can discover workers' event patterns and their accuracy for each video (R.2). The second type of information is overall annotations and event patterns of multiple tasks for each worker. This type allows analysts to reveal each worker's event patterns and the sensitivity of his/her annotations for

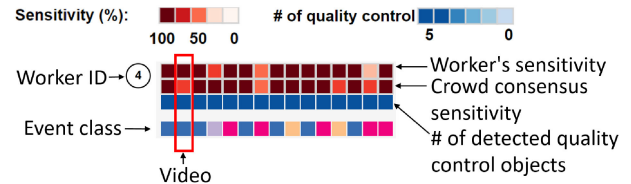


Fig. 8. Illustration of our worker view for a worker. For each video the worker annotated, we display the sensitivity of a worker (1st row), the crowd consensus sensitivity (2nd row), the number of quality control objects detected by the worker (3rd row), and the worker's event class (4th row).

multiple tasks (R.3). The Worker View for each worker can order workers by the date they completed the first task, by the averaged accuracy of tasks they completed, and by the number of completed tasks. Similar to our Timeline View, the Worker View for each video and for each worker consists of two parts: a summary of the result of each worker and details of the result. We note that the Worker View for each video is one of our main views, and the Worker View for each worker is used when we need to explore each worker's behavior in details.

Worker View for Each Video. The right side shows each annotation by a worker at the top and the worker's logged events at the bottom (Fig. 7). More specifically, at the top of the view, each box indicates an annotation, and a filled rectangle inside the box represents the area and position of the annotation. The outside box has the same aspect ratio as the input video frame. The color of the outside box represents the type of the worker's annotation (true positive or false positive). At the bottom of the right side, we visualize a worker's logged events by using red upside-down triangles (rewinding) and horizontal bars, where the color of each bar indicates the event type (e.g., playing and drawing) and the width of each bar shows the duration of each event.

On the left side of the view, we place squares to display the number of identified and missed target objects by a worker and the size of the corresponding target objects, in the same way as the Timeline View. Additionally, we show the number of detected quality control objects beneath the bars for the sizes of target objects. A circle with each worker's ID is colored to indicate the class of a worker's logged events. The color is assigned by the result of clustering workers' logged events in our data analysis component. This color for each event class is used for the same class in other views. When an analyst selects an annotation, we highlight the corresponding frame in the Timeline View and the corresponding annotation in the Frame View.

Worker View for Each Worker. We visualize overall information regarding annotations from the same worker in a single group by displaying a worker ID with a matrix visualization (Fig. 8). In the matrix visualization with three rows, each column indicates a video that the worker completed. The color of the first and the second rows show the sensitivity of a worker and the overall sensitivity of the crowd consensus for each video, respectively. The color of the third row represents the number of quality control objects detected by the worker. Below this, a fourth row displays the event class of the worker for each completed videos, where the same colors used in the Worker View for each video are assigned for the event class.

When an analyst selects a square in the view, we highlight all squares in the same column and focus on the corresponding video in the Timeline View. When an analyst hovers over a square, it shows brief information about the corresponding video as a tooltip.

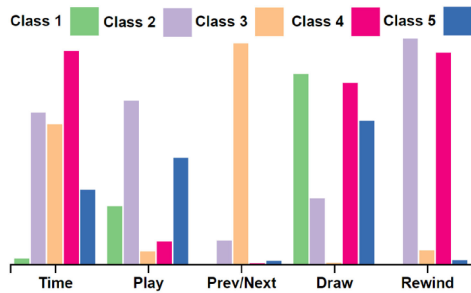


Fig. 9. Our class view shows the distribution of five events for the five clustered event classes. Workers belonging to class 4 viewed the videos by rarely using the buttons to step to the previous and next frame.

5.4 Class View

In the Worker View, we assign a color to each event class by clustering the workers' events. An analyst cannot understand the characteristics of each class without any description, which is essential for discovering the correlation between the workers' event patterns and the sensitivity of their answers (R.4). To show the characteristics of each class, we provide the Class View (Fig. 9). In the Class View, we use a grouped bar chart, where each group shows each event used for the clustering. Within each group, we visualize the averaged values of the event in each class.

5.5 Matrix View

Our Worker View shows the correlation between event classes and the sensitivity of annotations for each video or each worker. To discover the correlation between event classes and the sensitivity of corresponding annotations for all videos (R.4), we provide the Matrix View (Fig. 10). In the Matrix View, each column indicates a different event class, and each row represents a predefined attribute of a target object, such as its size or the number of frames it spans. The color of each cell in the view illustrates the sensitivity of the corresponding column and row.

5.6 Video View

Other views focus on the workers' and ground truth annotations. However, in cases such as VC videos, workers view a video where a virtual camera moves quickly at sharp bends along the navigation path and slowly when the navigation path is straight. Thus, understanding the context of an annotation in the video is useful to analyze missed target objects and false positive objects (R.2). For this purpose, we provide the Video View (Fig. 1 F), where an analyst views a selected video.

6 CASE STUDIES

We demonstrate the effectiveness of CMed with two datasets: polyp detection in VC videos and lung nodule detection in CT thin-slab MIP videos. We interviewed a radiologist for feedback on our CMed platform to gain more insights. A supplementary video demonstrates how our framework works interactively.

6.1 Virtual Colonoscopy Datasets

This study utilized VC data from 14 patients, yielding a total of 56 VC videos (antegrade and retrograde directions in both supine and prone scans) and thus a total of 56 unique tasks (HITs) (Figs. 1, 4). These 14 patients contained a total of 33 polyps, with the number of polyps per patient ranging from 1 to 5. Each VC video was viewed by ten workers,

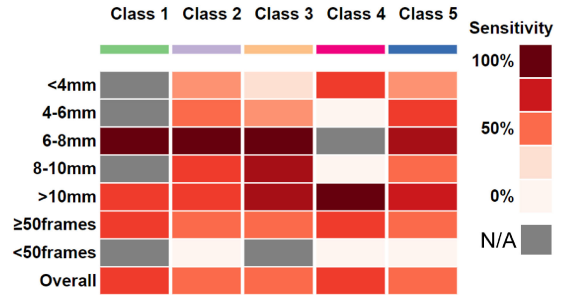


Fig. 10. In the matrix view, each column indicates a different event class, and each row represents a predefined attribute of a target object. The color of each cell in the view illustrates the sensitivity of the corresponding column and row.

except for two which were viewed by eleven workers, and a total of 125 workers participated.

We sorted the videos based on the polyp identification sensitivity. We then chose one video where most workers identified a polyp, but worker 3 missed the polyp (R.1). In order to identify whether or not he is an ugly worker, we explored his task history and event logs (R.3). We found that in some tasks he found all polyps, and thus he is not an ugly worker.

We also analyzed the event patterns of workers who completed multiple tasks (R.3). We first ordered workers by the number of tasks they completed. We then inspected workers who completed multiple tasks, and found that their performance and sensitivity did not improve over time. One interesting finding was that some workers who completed multiple tasks detected all quality control objects in some cases, but their sensitivity was lower than the sensitivity of the crowd consensus for those cases. However, in other cases, they had similar sensitivity to the crowd consensus even when they missed one or two quality control objects. Missing quality control objects does not necessarily mean a worker is a bad or ugly worker.

Next, we ordered workers by the averaged sensitivity of the tasks they completed. We found that worker 4 overall was as good as or better than other workers in tasks he completed (Fig. 8) (R.3). He completed 15 videos and found all polyps in 12 videos. In one task, he missed two polyps, but he marked several items which looked like polyps but were actually not polyps, such as segmentation artifacts (R.2). Thus, he is not an ugly worker and he might need additional training cases to improve his sensitivity.

We also looked at a worker with overall low sensitivity. Worker 57 detected all quality control objects, but he missed a polyp (R.3). We explored his annotations and found that he marked several objects which looked like polyps, but were actually not polyps (R.2). Thus, he also needs additional training cases to improve his sensitivity.

Additionally, we explored workers' annotations in the Timeline View and found that there were regions where several workers marked some objects as polyps (Fig. 11) (R.1). However, those were not actually polyps, but objects which looked like polyps, such as segmentation artifacts. These objects can be added as false positive cases in a training section to improve a worker's ability to discriminate between actual polyps and polyp-like objects. We also explored polyps that appear in only a few frames of the video (R.1). Among these polyps, some were detected by the crowd workers, but others were not. We chose one of these polyps and found that not only did this polyp appear in just a few frames, but it also looked like part of a fold. For future applications, we can add

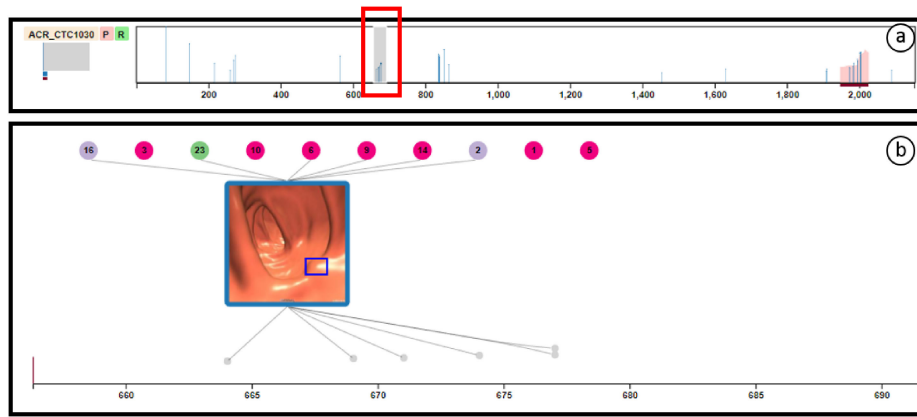


Fig. 11. Several workers marked an object as a polyp (a red box in (a)). However, that was actually not a polyp, but an object that looked like a polyp. (b) This case can be added as false positive case in a training section to improve users ability to discriminate between actual polyps and polyp-like objects.

this case into a training section and reduce the playback speed of videos.

In analyzing the Class View, class 1 was the overall best for all categories of polyps (R.4). Not surprisingly, workers in class 1 made an increased effort to detect polyps by clicking the buttons to step to the previous and/or next frame and used the rewind button more frequently. They also spent more time on completing the tasks, as compared to workers in other classes.

6.2 Lung CT Datasets

We used 15 videos. The videos contained a total of 169 nodules with the number of lung nodules per video ranging from 1 to 10. Each video was viewed by ten workers, and a total of 30 workers participated.

In the Timeline View, we explored several lung nodules that were missed by most workers (R.1). These nodules can be added as false negative cases in a training section. Additionally, similar to the VC datasets, we found that there were regions where several workers marked some objects as lung nodules that were not actually lung nodules (Fig. 12). These objects can be added as false positive cases in a training section.

We ordered workers by the averaged sensitivity of the tasks. We found a worker who completed three videos, but only detected the gorilla in each video and didn't annotate

anything else (Fig. 13) (R.3). The worker viewed the videos several times by rewinding them (R.2). He also marked several areas, but in the end deleted all annotations. Thus, he might be a bad worker and require more training cases.

Most of the workers were in class 1 (R.4). One interesting observation is that a few workers who were in class 4 detected all lung nodules which were greater than 10 mm (Fig. 10). These workers in class 4 spent more time on completing the tasks and used the rewind button more frequently than workers in other classes, but they rarely used the buttons to step to the previous and next frame (Fig. 9). Thus, to detect large lung nodules (> 10 mm), workers might need only the play and pause buttons.

6.3 Expert Feedback

In addition to conducting two case studies, we also demonstrated our system to two analysts (A1, A2) who analyze various data, including using/generating annotations for machine learning, and a radiologist (R) trained for VC and lung nodule detection who will use the output of the system. On a scale of 1 (novice: no experience) to 5 (expert: authoritative understanding and experience), the level of expertise in interpreting medical data for A1, A2, and R were 2, 3, and 5, respectively. First, we showed the different interactive visual components of our platform to them. After demonstrating each view of our CMed framework, they gave feedback on our CMed tool by putting

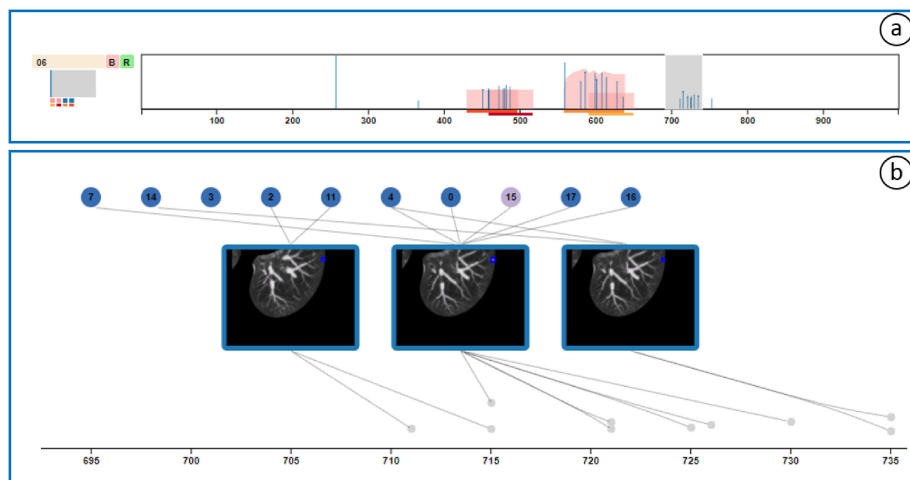


Fig. 12. Several workers marked some objects as lung nodules in the Timeline View. (a) However, in the Frame View. (b) We found that those were not actually nodules, but objects which looked like lung nodules.

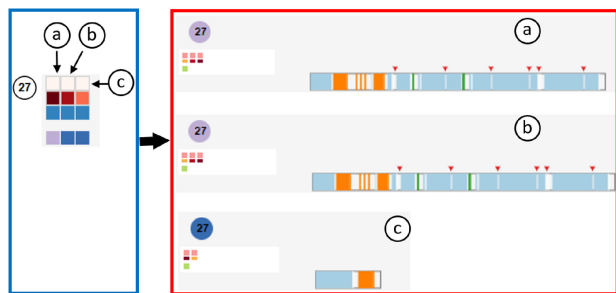


Fig. 13. In the worker view for each worker (a blue box), we found that a worker didn't annotate anything but only detected a gorilla (blue box) in three videos (a-c). In the corresponding Worker View for each video (a red box), the worker viewed videos ((a) and (b)) several times by rewinding them. He/she also marked several areas, but in the end deleted all annotations ((a-c) in a red box).

forward different hypotheses which we tested using the above case studies. With a brief explanation, they were able to gain insights from each view.

In general, the feedback for CMed was positive in terms of classifying workers based on their behavior patterns and finding workers' behavior patterns, what workers annotated, and the accuracy of spammers and non-spammers. *A2* and *R* especially liked our Frame View because it showed what workers annotated or marked through an image with a bounding box. *A1* and *R* found that the Worker View for each video was interesting because it showed the details of worker behavior patterns, such as how much a worker used the rewind functionality and how long the workers spent on each video. *A2* and *R* mentioned that insight gained from the Worker View from each worker was interesting because they could observe changes in the worker's behavior pattern even though they were still good workers. *R* said, "this finding makes sense because some radiologists spend a tenth of another radiologist's time and find the major findings". He expressed that based on this insight, he can rely on someone who had good accuracy on his/her prior tasks.

A1 and *A2* liked our linked views because they can interactively navigate information. They also expressed that our framework would greatly reduce their time to generate annotation. They would like to use our framework for other domains such as scientific data, which is also a time-consuming task for domain scientists. *R* also liked the Matrix View because it provided him with a summary of workers, stating that "For VC, sizable polyps can fairly reliably be detected by someone with no training". He expressed that based on overall insights obtained from our CMed system, he could use workers' results as a second reader, similar to a computer-aided detection algorithm. Lastly, all participants suggested that dynamic view configuration could be helpful since not all views are required for different types of analyses (even though each view provides complementary information).

7 GUIDELINES

In this section, we first describe design guidelines/considerations for crowdsourcing applications and visual analytics for medical crowdsourcing applications. We then suggest how to integrate the output of CMed into a clinical workflow.

7.1 Design Guidelines/Considerations

We have learned several lessons from our framework to help design future crowdsourcing applications and visual analytics for medical crowdsourcing applications as follows:

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 18, 2024 at 17:53:35 UTC from IEEE Xplore. Restrictions apply.

Quality Control. Some workers miss some quality control objects, but have good accuracy since they focused on the target objects. Thus, we cannot ignore workers who miss a few quality control objects. It would be interesting to add a visual analytics component with (semi) automatic filtering of ugly workers.

Tutorial/Training Examples. Additional training and more examples of tricky polyps/lung nodules might help improve the ability of the crowd in identifying polyps which look like folds. For this purpose, we can cluster annotations from other workers and visualize some of the representative images as training examples.

Workers Expertise. Since some workers who completed many tasks may perform well consistently, and understand better than other workers, their answers may be more reliable than those of other workers. Thus, we can rank workers (which should be regularly updated to prevent turning a good worker to an ugly worker), and consider the expertise level of a worker when compiling workers' answers.

Different Angles of Data. For VC dataset, some polyps are only visible from a certain fly-through. If we register different fly-throughs and visualize corresponding parts in each video, we can further improve the output of the framework.

7.2 Integration into Clinical Workflow

During this study and our previous studies, we held regular discussions with radiologists regarding how to integrate crowd annotations into a clinical workflow. The result of our framework can be integrated into a clinical workflow as a second reader or as annotations for deep learning. As a second reader, the clinical workflow is as follows: (1) patient CT data is acquired, (2) VC flythrough videos are generated and uploaded to the crowdsourcing platform, (3) crowd annotations are collected within 5 days, (4) analysts verify the annotations via CMed, and finally (5) the radiologist first performs VC inspection independently and then checks the verified crowd annotations to confirm the diagnosis (second reading). For this workflow, the radiologists we interviewed were willing to bear the cost of prior crowd interpretation by letting go of a meager fraction of their compensation, for convenience and for a stronger corroboration of their final diagnosis.

Recently, deep learning approaches have been shown great success in many image processing areas such as image classification [35] and image registration [36]. However, deep learning approaches require a large amount of training data/images for decent performance/accuracy. Existing detection methods in the medical domain have decent performance (e.g., >90% sensitivity for polyp detection [37], [38]), but require a large amount of training data/images to improve performance/accuracy. Collecting annotated data/images is time-consuming and expensive, especially in a medical domain. The possibility of using crowdsourced annotation data for a detection task in medical images has been shown [39], and thus the output of our framework could be used for training data. As deep learning approaches improve using the crowdsourced annotations, these approaches could replace the crowd as a second reader.

8 DISCUSSION

Our case studies and expert feedback demonstrate the effectiveness and efficiency of our CMed framework. However, there are several limitations in our current framework.

Scalability. Our system follows Keim's mantra of "Analyze first, show the important, zoom, filter and analyze further,

details on demand.” [40]. The Timeline View and the Worker View for each worker allow a crowdsourcing analyst to analyze the workers’ annotations and their behavior patterns. When the analyst orders videos/workers by a provided criteria and selects a region or a video in these views, the Frame View and the Worker View for each video show the details of the selected region or video. When a video is selected, we show all of the workers’ annotation information for the selected video. However, a previous study has shown that 10-20 workers per video are sufficient to obtain high quality output for medical data [41]. In our previous studies [22], [23], we could achieve high sensitivity and specificity by collecting annotations from 10 workers per video. However, if a hundred workers annotate the same video, the crowdsourcing application analyst can scroll down the Worker View to view all of the workers’ annotations.

Generalizability. In our approach, the size of each target object is encoded as a color. Thus, we can cater for up to 12 different sizes/types of target objects [42]. In some other datasets, there can be more categorized sizes/types. However, this number of colors is enough for most of our potential target applications, such as detecting cysts in virtual pancreatography.

Clustering. Our current clustering algorithm determines whether two annotations belong to the same object if they are in the same frame or in the closest frames. Based on our experience, most objects are not occluded by other objects in our potential applications, such as videos from medical images (CT or MRI). However, in VC 3D flythrough videos, a possible sequence of events is as follows: a polyp appears, becomes occluded by a fold, and then appears again. In this case, we cannot cluster the annotations corresponding this polyp into the same group. Since the purpose of our current clustering algorithm was just to minimize the total number of annotations, we did not address this possibility. We will investigate a better approach for enforcing this grouping of annotations as part of our future work.

Color Encoding. Even though our potential target users (crowdsourcing analysts and medical experts) liked our current color schemes, these might be cumbersome for some users. We used mainly two color encoding schemes to depict different annotation types and event types. Additionally, each clustered event class was depicted as another color scheme. A color scheme for the clustered event class can be simplified (the same color for all classes) when the Class View and Matrix View are hidden, which are not our main views. An alternative coloring scheme for the clustered event class is deploying a semantic color scheme based on the correlation between event classes and the sensitivity of annotations (e.g., a class with positive correlation: blue, a class with negative correlation: red). Instead of using colors for depicting the size of target objects, we can change only the lightness of a color to reduce the number of colors remembered by users.

9 CONCLUSION AND FUTURE WORK

We presented CMed, a novel visual analytics system for the interactive exploration of medical data annotations. We defined design requirements based on previous crowdsourcing studies. Our designed framework provides various insights into crowdsourced clinical data, which cannot be provided by other tools. Our case studies demonstrate the usefulness and effectiveness of our framework. Thus, CMed can help crowdsourcing application analysts/developers to design better crowdsourcing studies.

In the future, we plan to use our framework for other crowdsourcing biomedical applications such as virtual pancreatography and microscopy imaging. In the current system, we only deal with logged events related to a video player and drawing a bounding box in an annotation tool. However, there can be different types of events such as scrolling up/down and browser focus changes [5], depending on the annotation tools. We will incorporate these events into our framework to see the effects of these events. Lastly, in our study, the event patterns we defined had no effect on a worker’s performance. However, there may be a correlation between an event pattern(s) including some other events such as switching tabs and a worker’s performance. We will deploy an interactive learning to detect such behaviors.

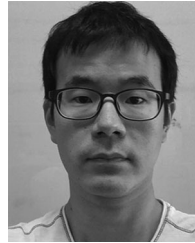
ACKNOWLEDGMENTS

The VC datasets are courtesy of Stony Brook University Hospital (SBUH) and Dr. Richard Choi, Walter Reed Army Medical Center. The lung datasets are courtesy of Lung Image Database Consortium. We would like to thank Dr. Matthew Barish and Dr. Kevin Baker of SBUH for their help in this project. This work has been partially supported by the US National Science Foundation Grants NRT1633299, CNS1650499, and OAC1919752, and the Marcus Foundation.

REFERENCES

- [1] S. Suri, D. G. Goldstein, and W. A. Mason, “Honesty in an online labor market,” in *Proc. Assoc. Advancement Artif. Intell. Human Comput. Workshop*, 2011, pp. 61–66.
- [2] L. Maier-Hein et al., “Crowd-algorithm collaboration for large-scale endoscopic image annotation with confidence,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2016, pp. 616–623.
- [3] E. Kamar, A. Kapoor, and E. Horvitz, “Identifying and accounting for task-dependent bias in crowdsourcing,” in *Proc. AAAI Conf. Human Comput. Crowdsourcing*, Aug. 2015, pp. 92–101.
- [4] G. Kazai, J. Kamps, and N. Milic-Frayling, “Worker types and personality traits in crowdsourcing relevance labels,” in *Proc. Conf. Inf. Knowl. Manage.*, 2011, pp. 1941–1944.
- [5] J. Rzeszutarski and A. Kittur, “CrowdScape: Interactively visualizing user behavior and output,” in *Proc. Symp. User Interface Softw. Technol.*, 2012, pp. 55–62.
- [6] A. Kachkaev, J. Wood, and J. Dykes, “Glyphs for exploring crowdsourced subjective survey classification,” *Comput. Graphics Forum*, vol. 33, no. 3, pp. 311–320, 2014.
- [7] J. H. Park, S. Nadeem, S. Mirhosseini, and A. Kaufman, “C2A: Crowd consensus analytics for virtual colonoscopy,” in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, Oct. 2016, pp. 21–30.
- [8] A. Kittur, E. H. Chi, and B. Suh, “Crowdsourcing user studies with mechanical turk,” in *Proc. Conf. Human Factors Comput. Syst.*, 2008, pp. 453–456.
- [9] C. Vondrick, D. Patterson, and D. Ramanan, “Efficiently scaling up crowdsourced video annotation,” *Int. J. Comput. Vis.*, vol. 101, no. 1, pp. 184–204, Jan. 2013.
- [10] A. Sheshadri and M. Lease, “SQUARE: A benchmark for research on computing crowd consensus,” in *Proc. AAAI Conf. Human Comput. Crowdsourcing*, 2013.
- [11] J. Whitehill, T. F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2035–2043.
- [12] M. T. McKenna, S. Wang, T. B. Nguyen, J. E. Burns, N. Petrick, and R. M. Summers, “Strategies for improved interpretation of computer-aided detections for CT colonography utilizing distributed human intelligence,” *Med. Image Anal.*, vol. 16, no. 6, pp. 1280–1292, 2012.
- [13] J. H. Park et al., “Crowdsourcing for identification of polyp-free segments in virtual colonoscopy videos,” *Proc. SPIE Med. Imag.*, vol. 10138, pp. 101 380V–101 380V–7, 2017.

- [14] P. Mettes and C. G. M. Snoek, "Pointly-supervised action localization," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 263–281, 2019. [Online]. Available: <https://doi.org/10.1007/s11263-018-1120-4>
- [15] Z. Yang, D. Mahajan, D. Ghadiyaram, R. Nevatia, and V. Ramanathan, "Activity driven weakly supervised object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2917–2926.
- [16] M. Romero, J. Summet, J. Stasko, and G. Abowd, "Viz-A-Vis: Toward visualizing video through computer vision," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1261–1268, Nov./Dec 2008.
- [17] M. L. Parry, P. A. Legg, D. H. Chung, I. W. Griffiths, and M. Chen, "Hierarchical event selection for video storyboards with a case study on snooker video visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 1747–1756, Dec. 2011.
- [18] A. H. Meghdadi and P. Irani, "Interactive exploration of surveillance video through action shot summarization and trajectory visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2119–2128, Dec. 2013.
- [19] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf, "Visual analytics for mobile eye tracking," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 301–310, Jan. 2017.
- [20] W. Willett, S. Ginosar, A. Steinitz, B. Hartmann, and M. Agrawala, "Identifying redundancy and exposing provenance in crowd-sourced data analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2198–2206, Dec. 2013.
- [21] S. Breslav, A. Khan, and K. Hornbæk, "Mimic: Visual analytics of online micro-interactions," in *Proc. Int. Work. Conf. Adv. Vis. Interfaces*, 2014, pp. 245–252.
- [22] S. Boorboor, S. Nadeem, J. H. Park, K. Baker, and A. Kaufman, "Crowdsourcing lung nodules detection and annotation," *Proc. SPIE Med. Imag.*, vol. 10579, 2018, Art. no. 105791D.
- [23] J. H. Park, S. Nadeem, J. Marino, K. Baker, M. Barish, and A. Kaufman, "Crowd-assisted polyp annotation of virtual colonoscopy videos," *Proc. SPIE Med. Imag.*, vol. 10579, 2018, Art. no. 105790M.
- [24] *Viatronix, V3D[®]-Colon*, <http://www.viatronix.com/ct-colonography.asp>.
- [25] J. W. Wallis, T. R. Miller, C. A. Lerner, and E. C. Kleerup, "Three-dimensional display in nuclear medicine," *IEEE Trans. Med. Imag.*, vol. 8, no. 4, pp. 297–230, Dec. 1989.
- [26] S. G. Armato et al., "The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans," *Med. Phys.*, vol. 38, no. 2, pp. 915–931, 2011.
- [27] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
- [28] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.
- [29] C. Kang, L. Zhu, X. Qian, J. Han, M. Wang, and Y. Y. Tang, "Geometry and topology preserving hashing for SIFT feature," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1563–1576, Jun. 2019.
- [30] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
- [31] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973.
- [32] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.
- [33] M. Harrower and C. A. Brewer, "ColorBrewer.org: An online tool for selecting colour schemes for maps," *Cartographic J.*, vol. 40, no. 1, pp. 27–37, 2003.
- [34] D. Holten and J. J. Van Wijk, "Force-directed edge bundling for graph visualization," *Comput. Graphics Forum*, vol. 28, no. 3, pp. 983–990, 2009.
- [35] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.
- [36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Med. Image Comput. Computer-Assisted Intervention*, vol. 9351, pp. 234–241, 2015.
- [37] Z. Yuan et al., "Automatic polyp detection in colonoscopy videos," *Proc. SPIE Med. Imag.*, vol. 10133, 2017, Art. no. 101332K.
- [38] P. N. Figueiredo, I. N. Figueiredo, L. Pinto, S. Kumar, Y.-H. R. Tsai, and A. V. Mamonov, "Polyp detection with computer-aided diagnosis in white light colonoscopy: Comparison of three different methods," *Endoscopy Int. Open*, vol. 07, no. 02, pp. E209–E215, 2019.
- [39] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab, "AggNet: Deep learning from crowds for mitosis detection in breast cancer histology images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1313–1321, May 2016.
- [40] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis," in *Proc. Int. Conf. Inf. Vis.*, Jul. 2006, pp. 9–16.
- [41] T. B. Nguyen et al., "Distributed human intelligence for colonic polyp classification in computer-aided detection for CT colonography," *Radiol.*, vol. 262, no. 3, pp. 824–833, 2012.
- [42] C. Ware, *Information Visualization: Perception for Design*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2012.



Ji Hwan Park received the PhD degree in computer science from Stony Brook University, in 2017. He is a research associate with Brookhaven National Laboratory. His research interests include visual analytics, information visualization, scientific visualization, machine learning, and human computer interaction.



Saad Nadeem received the PhD degree in computer science from Stony Brook University, in 2017. He is a research scholar with the Department of Medical Physics, Memorial Sloan Kettering Cancer Center. His research interests include medical imaging, bioinformatics, computer vision, computer graphics, and visualization.



Saeed Boorboor received the BSc (honors) degree in computer science from the School of Science and Engineering, Lahore University of Management Sciences, Pakistan. He is working toward the PhD degree with the Department of Computer Science. His research interests include computer vision, computer graphics, and visualization.



Joseph Marino received the PhD degree in computer science from Stony Brook University, in 2012. He is a postdoctoral associate with the Department of Computer Science, Stony Brook University. His research interests include visualization, medical imaging, and computer graphics.



Arie Kaufman received the PhD degree in computer science from BenGurion University, Israel, in 1977. He is a distinguished professor, director of the Center for Visual Computing (CVC), and chief scientist of the Center of Excellence in Wireless and Information Technology (CEWIT), Stony Brook University. He is internationally recognized for his pioneering and seminal contributions to visualization, graphics, virtual reality, and their applications, especially in biomedicine. He is fellow of the National Academy of Inventors, fellow of the IEEE, fellow of the ACM, member of the European Academy of Sciences, recipient of IEEE Visualization Career Award, and was inducted into LI Technology Hall of Fame. He was the founding editor-in-chief of the *IEEE Transaction on Visualization and Computer Graphics*, 1995–1998.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.