Covfee: an extensible web framework for continuous-time annotation of human behavior

Jose Vargas-Quiros J.D.VARGASQUIROS@TUDELFT.NL
Stephanie Tan S.TAN-1@TUDELFT.NL
Chirag Raman C.A.RAMAN@TUDELFT.NL
Laura Cabrera-Quiros L.C.CABRERAQUIROS@TUDELFT.NL
Hayley Hung H.HUNG@TUDELFT.NL

Abstract

Continuous-time annotation, where subjects annotate data while watching the continuous media (video, audio, or time series in general) has traditionally been applied to the annotation of continuous-value variables like arousal and valence in Affective Computing. On the other hand, machine perception tasks are most often annotated using frame-wise techniques. For actions, annotators find the start and end frame of the action of interest using a graphical interface. However, given the duration of the videos that are generally annotated in social interaction datasets, this can be a slow and frustrating process. It usually involves pausing the video at the onset or offset of the action and scrolling back and forth to identify the precise moment. A continuous annotation system, where annotators are asked to press a key when they perceive the target action to be occurring, can improve the time to do such annotations, especially in situations where single subjects are annotated for long periods of time. Keypoint annotations, where the task is to follow a particular point of interest in a video (e.g., a body joint) can also be done continuously. In this paper we present the Covfee web framework, a software package designed to support online continuous annotation tasks, with crowd-sourcing capabilities. We present results from case studies of continuous annotation of body poses (keypoints) and speaking (action) on an in-the-wild social interaction dataset. In the case of keypoints, we present a new technique allowing an easy way to follow a keypoint in a video using the mouse cursor. We found the technique to significantly reduce annotation times with no adverse effect on inter-annotator agreement. For action annotation, we used continuous annotation techniques to obtain binary speaking status labels and annotator ratings of confidence on those labels. Covfee is free software, available as a Python package documented at josedvq.github.io/covfee.

Keywords: annotation tool, human behavior annotation, continuous annotation, action annotation, pose annotation, crowd-sourcing

1. Introduction

Annotating human behavior for machine perception tasks involves the extraction of fine grained facial and body behaviour. Depending on the tasks or research questions being investigated, annotations may, for example, look to describe the movement and spatial location of a person via bounding box or keypoint annotations, indicate what actions are being performed by such person via binary action annotations, or describe the state of the person by annotating constructs like enjoyment or involvement.

Clearly, not all machine perception tasks and associated annotation tasks are created equal. Importantly, datasets containing human behavior vary widely in the number of subjects present in the dataset and the length of time each subject is recorded.

For example, most benchmarks in computer vision tasks of action recognition and pose estimation use still images or short video clips for training and benchmarking (Carreira and Zisserman, 2017). This often means there is a large number of data subjects in different environments, each recorded (and annotated) for a short period of time. This is desirable when the goal is to maximize the variability in the dataset to enhance the robustness of the system. In these tasks, annotations for keypoints are performed on individual frames, and videos are labeled with a single action.

In contrast, in applied machine learning within the social signal processing research, interacting subjects in audiovisual datasets need to be tracked and annotated for periods ranging from a few minutes to several hours (Cabrera-Quiros et al., 2018; Alameda-Pineda et al., 2016; Carletta et al., 2006) which is necessary to capture and study social interaction dynamics. Similarly in the affective computing community, datasets often involve annotating interactions lasting one hour or longer (McKeown et al., 2015). Thus, datasets for behavior analysis often have less data subjects, recorded for longer periods of time. Other applied fields working with in-the-wild data like surveillance and sports action recognition often require tracking subjects for long periods of time (Oh et al., 2011).

The annotation challenge is compounded when dataset are acquired in the wild (ie. without the benefits of lab-based, highly instrumented recording spaces), meaning that automatic techniques for subject detection and tracking and pose estimation are not applicable. Obtaining the same level of detail of human behaviour in these settings is often prohibitive in terms of the manual labour, or equivalently financial cost involved.

A second key characteristic of many human behavior annotations, especially those of actions and higher-order constructs, is the central role of temporal context in perception. While *simple* tasks such as the annotation of body joints in a video can be considered free of temporal context (ie. a single frame can be meaningfully annotated), annotating concepts which require a judgement about intention, such as *the use of sarcasm* or *dominant laughter* requires a judgement that can only be done with access to temporal context (ie. the past) of the interaction.

Annotating human subjects for long periods of time while having access to temporal context has created a need in annotation tooling that we argue is not covered by existing annotation tools and techniques. In this chapter we present a software framework offering a technical solution to this problem.

Continuous-time annotation refers to annotations being carried out in real time while the target media is being watched without pause. Traditionally, continuous-time annotation has almost-exclusively been applied to annotation of affect of a target subject, usually being observed in a video. Affect has been annotated via the variables in the circumplex model of affect: arousal and valence (considered continuous variables). Joint annotation of both variables was first proposed, where the annotator controls the position of a cursor within a labeled diagram (2D annotation) using their mouse (Cowie et al., 2000). Further developments split the annotation process into the separate annotation of arousal and valence (Cowie et al., 2013). Since then, continuous-time annotation has been used to annotate multiple datasets (Ringeval et al., 2017; Sharma et al., 2019), more modern tools have been developed (Girard and Aidan, 2018; Girard, 2018; Melhart et al., 2019), and the best way to make use of continuous-value annotations taking into account human biases has been re-

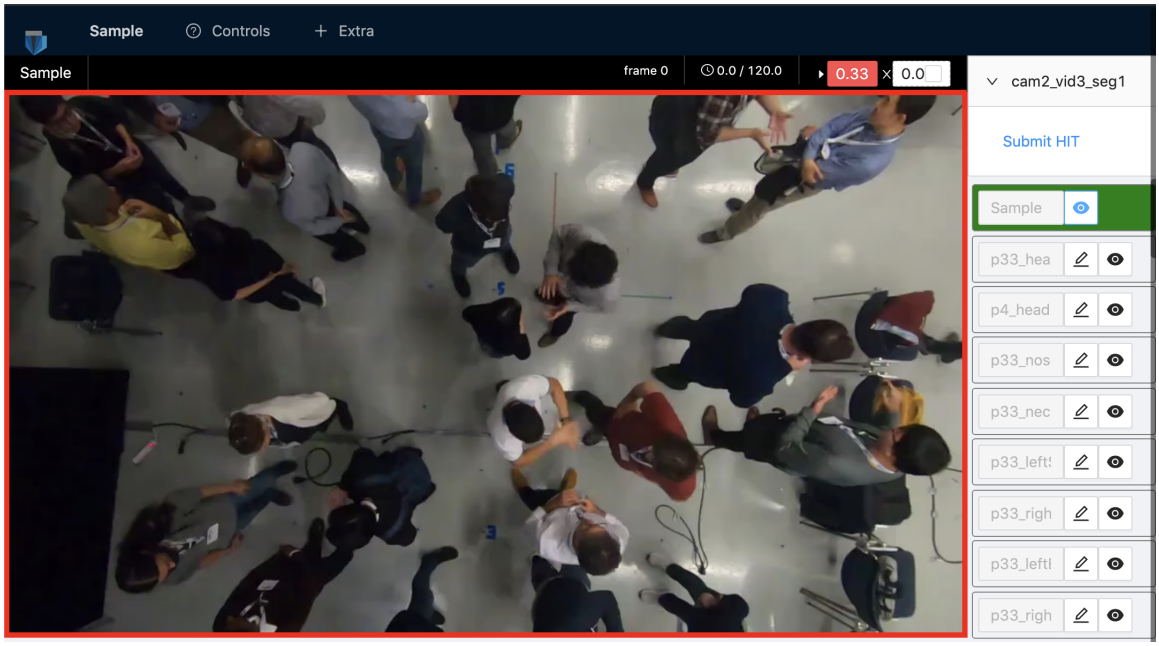

Figure 1: The Covfee keypoints annotation interface.

searched (Lopes et al., 2018; Booth and Narayanan, 2020); all within the context of affective computing.

In this chapter, however, we treat continuous-time annotation as a general technique applicable to different types of variables and different types of media. In addition to continuous affect annotation, examples of continuous-time annotation include holding down a keyboard key to indicate that a person in a video is speaking, following a person's hand with the mouse cursor to indicate its position, or controlling a continuous slider using the mouse to rate the perceived level of engagement of a person in an interaction.

We investigate the power of continuous-time annotation to improve annotation time in the labelling of human subjects for long periods of time. The continuous nature of this kind of annotation has the additional advantage of facilitating the perception of temporal context, potentially improving the quality of annotations through better annotator judgement of the target action or construct.

Annotating subjects for long periods of time can be made more feasible in practice by leveraging crowd-sourcing, splitting the load among multiple annotators. In crowd-sourcing, remote workers are paid to perform HITs (human intelligence tasks) consisting in units of work to be completed by one annotator (usually taking a few minutes to complete). Given its use in different fields, and notably computer vision and human behavior analysis fields, we leveraged its benefits by giving our continuous annotation framework crowd-sourcing capabilities.

Covfee, our software solution brings together the possibilities of continuous annotation with those of crowd-sourcing into a web-based annotation framework. Because annotation techniques are often task-specific and continuous annotation is a nascent field in need of

experimentation, we designed and documented Covfee as an extensible framework, with the goal of letting users implement new techniques with as little effort as possible. Particular continuous annotation techniques such as those for affect, action annotation or keypoint annotation are applications of the framework.

Our contributions are the following:

- We present Covfee, an open source web annotation framework with crowd-sourcing support, implementing continuous action and keypoint annotation out-of-the-box. Covfee supports the implementation of custom continuous annotation tasks with different media types and user interfaces taking advantage of existing capabilities for data serialization/storage, crowd-sourcing, qualification testing and annotation tracking and monitoring. Annotation tasks to be implemented in Covfee may range from existing techniques for rating of continuous variables like affective dimensions (Lopes et al., 2018) to novel techniques for vision tasks such as the ones presented in this paper.

- We present a case study involving the use of Covfee for the annotation of human body joints in an in-the-wild dataset. We present comparative results against a traditional annotation method and found a nearly three-fold improvement in annotation time with no loss in inter-annotator agreement.

- We present a second application of Covfee for the efficient annotation of actions in the same social interaction dataset. We analyze annotations of speaking status via a continuous binary interface and confidence ratings for those annotations.

- We discuss the advantages and disadvantages of applying continuous annotation to human behavior datasets. Based on both case studies, we provide recommendations and discussion of other potential use cases for Covfee.

The remainder of the chapter is organized as follows. In section 2 we start with a summary of related work and its relevance to the Covfee framework and its tasks. In section 3 we present the Covfee framework, summarize our main design requirements and decisions and present its main features for both basics users looking to annotate data or advanced users looking to implement new tasks on using Covfee. In section 4 we present two case studies using the Covfee framework for new types of continuous annotation: the fully manual annotation of human body joints using the mouse as tracking device; and the binary annotation of actions in a social scene. We end by discussing and reflecting on these case studies and the role of continuous techniques in human behavior annotation in Section 5.

2. Related work

In this section we start by reviewing work on manual annotation tools, with a focus on web computer vision and time series annotation tools. We go on to review work specific to continuous annotation, most of which relates to annotation of human subjects.

2.1. Manually annotating keypoints and actions

Of particular importance in computer vision tasks involving human data subjects are the tasks of pose estimation (Cao et al., 2017; Joo et al., 2016), or keypoint estimation in general, and the task of action recognition or localization (Carreira and Zisserman, 2017), both of which we address in this chapter.

Keypoint annotation involves the labelling of important points in an object of interest. This could be hand joints, facial landmarks, or object keypoints. Keypoint annotation is supported in tools like Vatic and CVAT via image-level annotations, performed every N frames, and interpolated in between. This is however a time-consuming process whose accuracy is limited by the interpolation step, particularly if a keypoint being tracked moves with highly varying levels of acceleration. The number of frames to skip should be few enough to avoid under-fitting a particular trajectory of a keypoint whilst still being large enough to minimise manual effort. It is also unclear how to deal with frequent occlusion of the target keypoint in this scenario and annotating for such occlusion makes the process slower.

As a result of these challenges, many works involving the tracking of many individuals in a social scene have reverted to using bounding boxes for subject localization despite the fact that full body poses contain richer information (Cabrera-Quiros et al., 2018). Others have reverted to using a much smaller set of skeletal points such as just using head positions and orientations (Alameda-Pineda et al., 2016). Parallel to this, there is a growing community working on the detection of actions directly from skeletal data (Gupta et al., 2020) given the emergence of large scale keypoint data that has been automatically generated in highly instrumented lab environments (Joo et al., 2016). Being able to annotate body keypoints in in-the-wild settings provides a sound basis for researchers in these areas to transition to working on more realistic natural settings.

The first step in action annotation involves the localization of actions of interest in a recording. In social interaction datasets such recordings often capture a large social event (Cabrera-Quiros et al., 2018), multiple meetings (Carletta et al., 2006) or conversations, spanning dozens of hours of individual interaction and requiring a time-consuming effort to annotate. To this end, actions are traditionally localized using a mouse and graphical interface. In tools such as ELAN (for Psycholinguistics, 2021), the user localizes the start and end frame of the action, which is then annotated by drawing an interval in a timeline. In tools such as Vatic and CVAT, actions are annotated via flags, which are turned on for the frame when the action is deemed to start, and off for the end of the action. Both of these approaches require the user to pause the video every time an action is recognized. This has the drawback of slowing down the process and making it harder for the annotators to follow the flow or dynamics of the interaction, or media in general.

An important consideration when annotating body keypoints is the annotation of occlusion: when the target body joint is not visible, due to being occluded by another object/person in the scene, or possibly the same person (self-occlusion). Rather than being constant, in many in-the-wild datasets, body keypoints may become visible and occluded frequently when bodies gesture, change posture or move around in the scene. Occlusion signals are important for training of pose estimation methods, and are included in several datasets for pose estimation (Lin et al., 2015; Andriluka et al., 2014). Networks designed to learn from

the occlusion signals have been shown to improve performance on pose estimation image datasets (Cheng et al., 2019; Zhou et al., 2020).

2.2. Crowdsourcing annotations

The advent of deep learning in computer vision has resulted in algorithms requiring large amounts of data to reach state-of-the-art performance. In video-based tasks like action classification (recognition) and localization, this has required the labeling of datasets with hundreds of hours of video, used now as benchmarking datasets. Improvements in the related task of pose estimation (Güler et al., 2018; Cao et al., 2017; Fang et al., 2017), commonly trained from images, has been possible thanks to image datasets with tens of thousands of examples (Andriluka et al., 2014; Lin et al., 2015). This process has come together with the development of annotation tools capable of supporting at least a range of canonical tasks: keypoint annotation, bounding box annotation, image segmentation and temporal (action) labeling.

With the move towards online services, crowd-sourcing annotations has gained relevance in computer vision. This trend led to the collection and annotation of large datasets completely online (Sigurdsson et al., 2016). In human behavior analysis in particular, crowd-sourcing has been used to annotate actions and person bounding boxes within the social signal processing community (Cabrera-Quiros et al., 2018) and more extensively used in the affective computive computing community (Barsoum et al., 2016; Busso et al., 2017; Lotfian and Busso, 2019), where techniques for improving reliability in the crowd-sourcing setting have been explored (Burmania et al., 2016).

In a comprehensive paper on the subject of crowd-sourcing Vondrick et al. (2013) use the Vatic tool to provide a series of insights related to online video annotations. Despite the fact that annotators used traditional frame-level annotation techniques, some of their insights are relevant to annotation processes and tools in generals, and we summarize them here. When annotating body joints, they found that annotators are more efficient and prefer to annotate one joint at a time throughout the whole video compared to annotating one image (all joints) at a time. Another important observation was that annotators "rely on the motion of objects in order to correctly decode the scene", and that "the user must watch the video play in order to correctly track [an object]" (Vondrick et al., 2013, p.7). Both of these are default choices in the continuous annotation paradigm, where the annotation technique must be simplified to be done while the video plays.

In the same paper, authors concluded that larger tasks, where a single annotator annotates all objects in a video are better than smaller tasks, such as different annotators annotating single objects. This is likely due to the overhead involved in familiarizing oneself with the scene to annotate. They also found that a constrained interface without too many choices will result in better annotation times, compared to more flexible ones. Authors address the importance of filtering workers through qualification tasks, stating that "because video annotation is hard, we found that most workers, despite accepting the task, do not have the necessary patience or skill to be accurate annotators." We take advantage of these important insights in the design of the Covfee framework and associated annotation techniques.

2.3. Continuous-time annotation

The term *continuous annotation* generally refers to *continuous-time* annotation. Although a precise delineation of what constitutes continuous-time annotation is not present in the literature, we will treat it as an umbrella term that describes the process of performing an annotation task while the target media is being watched (possibly in real time), usually without any pauses. A distinction must be made from continuous-value annotation which refers to the annotation of continuous variables in general which could also be carried out as a post-hoc annotation step. Although mostly applied to audiovisual recordings, continuous annotation is not limited to this set of modalities and may apply to any sensory experience such as listening to an audio recording or watching a live performance.

Continuous-time annotation started with the continuous recording of emotional states with Feeltrace, an instrument designed to let observers *track the emotional content of a stimulus as they perceive it over time* (Cowie et al., 2000, p.1). The interface consists of a circle, with dimensions corresponding roughly to arousal and valence (Russell, 1980), the dimensions in the widely-used circumplex model of affect (Russell, 1980; Posner et al., 2005). This type of continuous annotation allowed observers to describe an emotional state by moving a pointer within the circle using their mouse. The newer GTrace technique (Cowie et al., 2013), presented as a "Feeltrace successor" supported one-dimensional annotations of valence and arousal with visual feedback markers on a desktop application.

Continuous annotation has since been used in the affective computing community for the annotation of datasets for affect through GTrace-type interfaces. In datasets like DEAP (Koelstra et al., 2012), SEMAINE (McKeown et al., 2012), RECOLA (Ringeval et al., 2013) and DECAF (Abadi et al., 2015), valence and arousal were annotated separately using a mouse-controlled graphical interface. More recently, datasets like SEWA (Ringeval et al., 2017) and CASE (Sharma et al., 2019) have moved to the use of joysticks for simultaneous annotation of arousal and valence. The reasons cited by Sharma et al. (2019) are that separate annotation of arousal and valence does not account for the relationship between them, and that "mouse-based annotation tools are generally less ergonomic than joysticks". However, Metallinou and Narayanan (2013) more precisely state that Feeltrace and GTrace require continuously pressing the mouse to annotate, which is tiring when annotating long videos. CARMA (Girard, 2018) and DARMA (Girard and Aidan, 2018) are other desktop tools for continuous affect annotation with mouse and joysticks respectively.

More recently, RankTrace (Lopes et al., 2018) addressed the problem that humans are bad at maintaining references of continuous values, which is supported by theories such as the adaptation level theory. This theory suggest that "humans cannot maintain a constant value about subjective notions; instead, their preferences are made on a pairwise comparison basis using an internal ordinal scale" (Lopes et al., 2018, p.1). Their interface instead captures unbounded annotations, which are then interpreted using their gradient. They showed that the gradient of the unbounded annotations was a better predictor of skin conductance (as a correlate of emotion) than the absolute value of the annotations. They performed annotations using a hardware wheel for input. The issue of interpreting continuous-valued annotations directly relates to the question of how to measure agreement between annotators. To this end, Booth and Narayanan (2020) designed an ordinal agreement measure for continuous-time, continuous-value annotations, based on the observation that annotators

approximately preserve rank ordering and capture trends (increasing or decreasing) when annotating continuous values. These findings may limit the utility of continuous-value annotations (since they cannot be reliably compared absolutely). It is however unclear to what extent they generalize to the annotation of less subjective variables.

A major drawback of the previously-mentioned tools is that they were only implemented as Windows applications, making them unusable in a crowd-sourcing setting, and therefore hard to scale for use in large datasets. Web-based applications, in contrast, offer a lower barrier to access for annotators, do not require the annotators to store a local copy of the annotated media, and may support the crowd-sourcing of annotations in online marketplaces. The data storage issue is an important one when the data to be annotated is considered privacy sensitive. Streaming data for annotation through a web interface mitigates intentional or unintentional data privacy violations such as forgetting to delete the raw data after it has been used for annotation. PAGAN (Melhart et al., 2019) is possibly the first web-based tool for continuous annotation, with support for GTrace and RankTrace, as well as binary annotations. PAGAN specializes on affect annotations and is not geared towards supporting the implementation of custom techniques.

Continuous-time annotation has some inherent delay due to the time that annotators take to react and process their perception. This could potentially impair the performance of systems that are developed to learn from such data. Some recent efforts have concentrated on the study of these delays and how they can be corrected in the context of affect annotation (Huang et al., 2015). Marioryad and Busso (2015) align annotations by maximizing the mutual information between annotations and expressive behaviors as captured by facial action units and speech features. Khorram et al. (2019) present a convolutional network capable of jointly aligning and predicting continuous emotion annotations via a time-shifted low-pass filter. Although these works show improvements in regression performance with respect to baselines without correction, it is unclear how much such correction methods can improve performance, as the true delays in the studied datasets are unknown. Furthermore, Marioryad and Busso (2015) found no significant differences when using a constant delay, compared to their data-driven approach. Although work on delays has been exclusively done in the context of continuous-valued affect annotations, and delays in annotation are likely task-specific due to different stimuli processing times, some degree of human delay is inherent to all kinds of continuous annotation. It seems pertinent for researchers developing machine perception systems trained with continuously annotated data to be mindful of the potential effects (if any) of delay.

In summary, annotation in computer vision and continuous annotation are two completely disjoint fields in the literature. The former has focused on image-level techniques aided by interpolation for the annotation of keypoints for pose estimation tags, and the use of binary flags for the annotation of actions, used in action localization tasks. It is however unknown how such annotation techniques compare to continuous ones in time efficiency and annotation quality, and to what extent their non-continuous nature affects annotations heavily dependent on temporal context. A reason for this is that continuous annotation literature has almost-exclusively focused on the subject of affect, which has resulted in very specific techniques, tools and insights for continuous-time annotation of continuous affect variables. This means there is little study of the phenomenon of continuous annotation in its more general form, which may involve different modalities, input devices, and interfaces.

The lack of software support for the implementation of continuous annotation tasks also limits broader study of this topic. To this end, we hope that Covfee lowers the entry level for more researchers to explore continuous annotation in more settings allowing for a better understanding of its potential.

3. The Covfee framework for continuous annotation

Covfee was born out of the need of an advanced framework for both annotating existing datasets and researching questions related to continuous annotation. The target user of Covfee is thus a researcher aiming to annotate a human behavior dataset, or to use or implement novel continuous annotation interfaces for research purposes. As such, Covfee was built with a set of main broad requirements:

- To be under an open source license and documented online. Covfee has been released under an MIT license, a permissive license enabling among others the copying, modification and redistribution of the software without limitations.
- To be easy to install, and launchable on a local web browser from a command line.
- To be deployable in a public server for online annotation. This is also necessary to support crowd-sourcing annotations in online marketplaces.
- To support large annotation processes consisting of hundreds of HITs, with each ranging from seconds to hours in length (of the target media).
- To implement client-server communication of annotations and storage on the server. Annotations should be buffered (to prevent data loss from network errors) and submitted to the server where Covfee is deployed, where they should be easy to download by the requester.
- That annotation techniques implemented in Covfee (eg. binary annotation of videos, video keypoint annotation) are easy to reuse and re-deploy.
- To support additional functionality that is useful in an online annotation process: requesting non-continuous feedback from annotators (eg. demographics, experience feedback, etc), requiring agreement to terms and conditions (eg. an EULA) before getting access to the data, and providing rich annotation instructions (images, videos, tooltips) for users.
- To support automated qualification tasks via implementation of a validation method in Python. Validation methods receive the annotations and return a boolean decision on whether the submission passes the qualification test or not.
- That new custom tasks can be implemented with a basic knowledge of Javascript / Typescript by implementing a class with a specific interface; much like writing a custom network in modern deep learning frameworks can be done by implementing methods of a subclass

- To run in most modern desktop browsers. We do not discard making covfee tasks usable in mobile devices in the future/ However, due to the additional implementation effort this would require, we decided to start with desktop browser support only. Note that since tasks in covfee may be custom and use any browser features or APIs, particular tasks may have more reduced compatibility than Covfee as a whole. Covfee should provide a way to test for browser compatibility and instruct annotators to use a compatible browser before they start working on a task.

To support these broad goals, we implemented Covfee as a Python package available in the Python Package Index. Once installed, Covfee's administration panel can be started in a browser from the command line. The main building blocks of Covfee are shown in Figure 2. The web application was implemented in Typescript as a one-page-application in the popular React web framework. The web server makes use of the Flask framework and a SQLite database for annotation storage.

3.1. The Covfee specification file

An important questions in the design of Covfee was how to let requestors describe the HITs to be created in Covfee (ie. how would a researcher use Covfee?). Existing online annotation tools let the requester create HITs using a graphical interface where media files can be uploaded and the variables to be annotated are specified. Each HIT maps to an interface with tools to support different annotation techniques (eg. drawing bounding boxes, keypoints, setting binary flags). An annotator is expected to navigate this rich interface to annotate the requested variables. Designing Covfee in this way would have several major drawbacks. First, for large annotation processes with hundreds of files to be annotated (each of which would map to a different HIT), specifying HITs using a graphical interface would be cumbersome for the requester. Second, having a single rich interface with tools and options for different annotation techniques is not desirable. Richer interfaces with many options were found by Vondrick et al. (2013) to lead to information overload for annotators. Ideally the annotation interface should only contain the tools and information necessary to complete the HIT.

To avoid these drawbacks, we designed Covfee to read a JSON (Javascript Object Notation) specification file describing the HITs to be created as input. Instead of uploading media files using a graphical interface, URLs to the media files to annotate are part of the specification. Using a file following a particular structure makes it easy for the requester to generate this file using the programming language of their preference, an advantage for large annotation projects. For smaller annotation projects the file can also be created by hand based on the examples in Covfee's documentation. Because the Covfee specification maps directly to a set of HITs, it also serves as a shareable record to help other researchers reproduce a particular annotation project.

The specification file follows a particular structure, which among other things includes:

- Project details, including a name and details of the contact person. This information is shown to annotators in case they run into an issue during the annotation process.
- A list of HITs forming part of the project. Each HIT can be reproduced multiple times via a `repeat` parameter in the specification, or using the Covfee interface. Every

instance of a HIT is mapped to a URL, meant to be visited by one annotator. A HIT in Covfee consists in a set of sub-tasks, of possibly different types. For example, a HIT may contain a keypoint annotation task and a binary action annotation task.

- For each HIT, a list of tasks comprising it. The specification of a task is different depending on its parameters. For example, the specification of a keypoint annotation task is different from that of an action annotation task. Each task in the specification maps to an annotation interface that is specific to its task type (ie. annotation technique). This minimizes information overload for annotators by giving them only the tools, options and instructions relevant to the task at hand.

An example of a Covfee specification file is shown in Appendix A of the supplementary material. Specification files are validated by covfee to ensure that they have the correct structure and valid property names and values. Friendly error messages are returned indicating the location and cause of any error within the structure. This makes it easy for the user to debug their specification and avoids potentially hard-to-trace errors due to mistakes in the specification. Appendix A shows an example of validation output from Covfee.

On the technical side, validation naturally requires a model or schema of what the specification should look like. The use of Typescript for the implementation of tasks in Covfee provides a natural way to do this. Typescript interfaces are used to specify the shape and parameters of each tasks's specification. Covfee internally translates these interfaces into JSON Schema (json-schema org, 2020), a vocabulary for the validation of JSON documents. These JSON schema are used by the covfee CLI to validate the JSON structures. Figure 2 shows a diagram of covfee's architecture, including this process.

3.2. Online workflow

Figure 3 diagrams the workflow in Covfee. The main participants are the requester (researcher) and the annotators. Annotators get access to a Covfee interface generated by the requester using the framework.

The workflow to be followed by the requester can be put into a sequence of steps:

1. The requester creates a Covfee specification file. Covfee's documentation was designed to help the requester create the specification of each task.

2. The requester runs Covfee to validate the specification and generate the Covfee HITs from it. If the requester made a mistake in the specification, friendly error messages are returned indicating why and where the specification is invalid. Once a valid specification is provided, the requester can now enter Covfee's admin panel and obtain anonymized links to each HIT. A CSV file with all the links can be downloaded to be uploaded to Amazon MTurk or otherwise shared with annotators.

3. The requester may keep track of the annotation process using the admin panel. At any time it is possible to download the raw annotations in JSON and CSV formats.

For more information on the use of Covfee, please refer to Covfee's online documentation (Vargas Quiros, 2021).

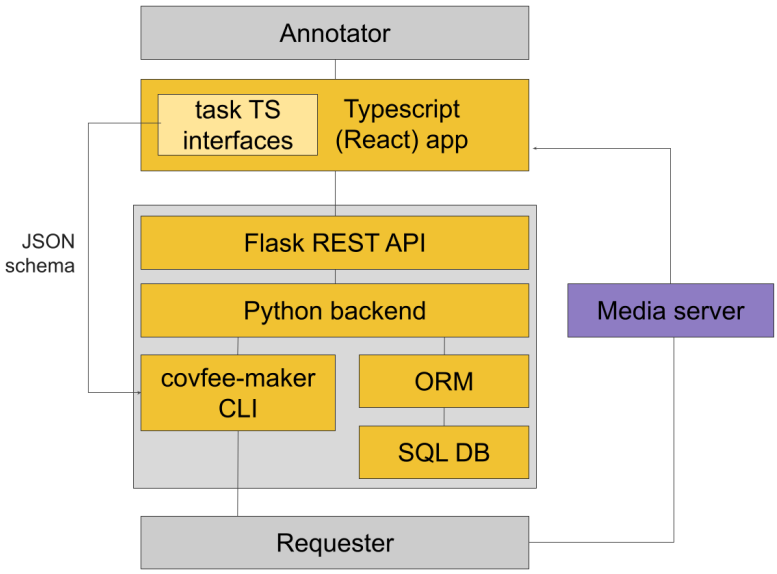

Figure 2: The architecture of the Covfee framework. A Python server with a relational database (SQL DB) and ORM layer (a layer mapping database tables into software objects) takes care of data storage and communication with the Typescript web application. The requester (researcher) interacts with Covfee via the *covfee make* CLI, which validates a user-provided specification of the HITs to be created. Typescript interfaces translated into JSON schema (a language for describing the structure of objects) are used as templates to validate the specification and provide friendly errors to the requester in case of mistakes in the specification.

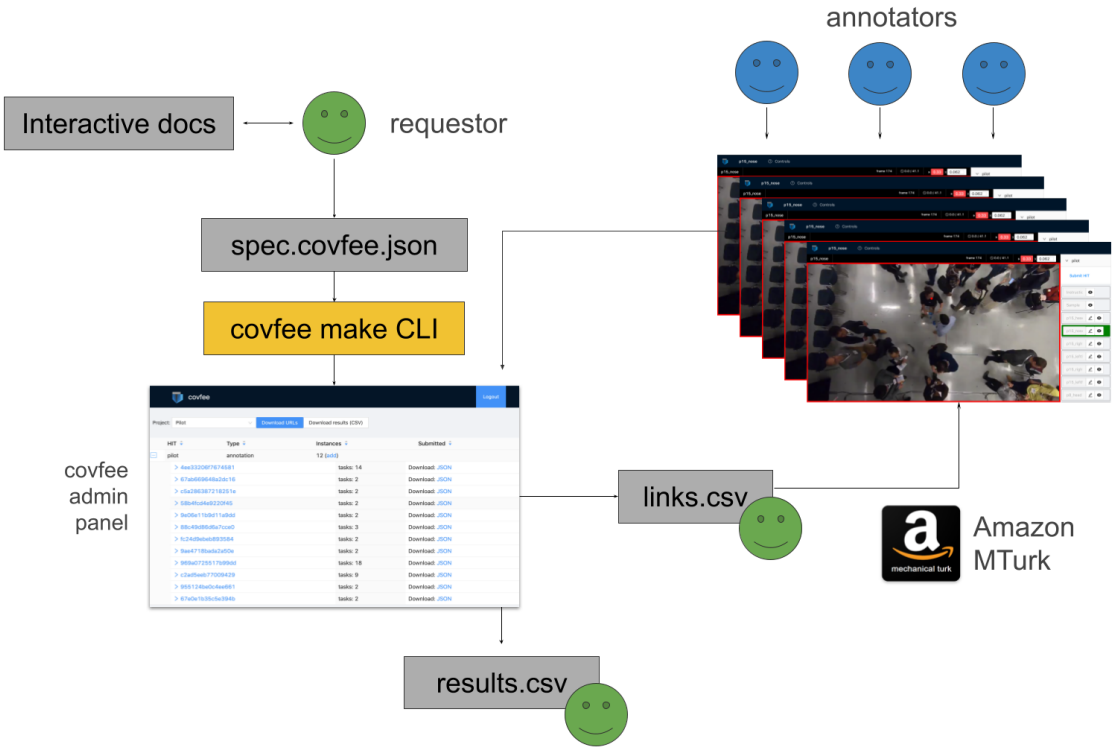

Figure 3: Covfee is designed to map a JSON specification into an online interface, meant to be replicated and shared online. In the basic workflow the requester uses the Covfee documentation to create the specification, which is used to generate the online HITs for remote annotators. The HIT URLs are accessible to the requester through the administrative panel.

3.3. Data privacy and security

Covfee deals with two kinds of potentially sensitive data: the dataset to be annotated (which could contain sensitive information about the data subjects); and the annotator responses, which could include personal information about annotators or data subjects.

Regarding the dataset, Covfee secures access to HITs via URLs containing a hash generated from a secret key. Hash URLs offer protection against scraping of the HIT links, resulting in unauthorized access to datasets while preserving the convenience of using URLs to share HITs. Using hash URLs is a standard practice for sharing documents online, with the drawback that any person with the URL may access the HIT. This is however an acceptable and somewhat necessary trade-off, given that annotators in crowd-sourcing platforms do not expect to need to create a user account in a third-party website to complete their task.

Covfee additionally provides support for data access control via required forms that must be filled in by the annotators before getting access to the data. Data access control is useful for datasets that are not publicly available in the internet, but require agreement with an End User License Agreement (EULA) on the part of the annotator. An EULA is put in place for these datasets to ensure that any person with access to the dataset agrees to the conditions stated in the agreement, which often include measures for protecting the privacy of data subjects. Many social interaction datasets are available only under an EULA (Cabrera-Quiros et al., 2018).

Regarding sensitive responses from annotators, consent elicitation is necessary under the European General Data Protection Regulation (GDPR) when the information requested from annotators includes non-optional sensitive information. Although the GDPR is European law, it applies to the handling of data from European citizens and residents regardless of location and is considered a global reference for data protection legislation. Covfee supports consent elicitation through the same mechanics of required forms, where the user must provide their consent before proceeding with the annotation process.

3.4. Crowd-sourcing support

Covfee was created with the goal of supporting the crowd-sourcing of tasks. In contrast with a non-crowd-sourced setting, where the annotators may often be given instructions in person or via video call, in the crowd-sourcing setting communication with the annotator is generally one-way. Annotators expect to be directed to a self-contained human intelligence task (HIT) to be completed normally in a few minutes, before returning to the crowd-sourcing platform. Maximizing information flow through clear, easy-to-follow instructions and means to obtain feedback from annotators are therefore key to support this setting.

Furthermore, crowd-sourcing platforms must interface with Covfee to validate the completion of a HIT. Here, we focused on supporting integration with a) Amazon Mechanical Turk (the most popular crowd-sourcing platform) and b) Prolific, a growing platform with a focus on research studies.

Important features in Covfee that make it possible to run crowd-sourced annotation flows efficiently are:

Support for rich instruction pages A special type of task (Instruction task) can be used to provide detailed instructions in Markdown/HTML (including video tutorials).

Additionally, any task in Covfee may contain tooltips to emphasize instructions or other information relevant for the annotator.

Questionnaire support Questionnaire tasks can be used to request non-continuous feedback from participants via free text boxes, buttons, sliders, and other static form elements.

Support for automatic qualification tasks For continuous tasks, a HIT may be opened only if the annotator demonstrates certain level of ability on a shorter qualification task. A usual qualification task consists in a short sample drawn from the dataset, on which annotators are asked to follow the annotation process to be followed on the full HIT. Covfee allows the requester to easily implement a validation method, typically to compute an error between the obtained annotations and some gold standard (eg. annotations performed by the requester), allowing for some level of discrepancy (typically set empirically). Qualification tasks have been shown to improve the quality of the annotations that can be obtained in major crowd-sourcing platforms platforms (Vondrick et al., 2013).

Completion codes and redirects Covfee implements integration between the Covfee platform and the crowd-souring platform via completion codes associated with each HIT. The completion code may be generated by covfee and provided to the requester (following the Amazon Mechanical Turk system) or manually provided by the requester (following the Prolific system). The completion code is shown to annotators on successful completion of their HIT, to be entered by them in the crowd-sourcing platform as proof of completion. Covfee also supports redirecting annotators to external URLs on completion of their HITs.

Admin panel The admin panel, only accessible by the requester, helps keep track of progress and allows easy bulk-download of HIT URLs for use in crowd-sourcing platforms.

3.5. Extensibility

Covfee achieves its role as a framework, rather than simply a tool, thanks to the task-oriented class design; a user can create new Covfee tasks easily by sub-classing an existing base class. Javascript objects are available for developers to interface with. For example:

Covfee takes care of annotation recording . Covfee has methods for submitting data to the server and for reading data back from it. Data storage and client-server communication are abstracted away by Covfee. Continuous annotations are timestamped, buffered and sent to the server in chunks to minimize the risk of data loss. In addition to continuous data, tasks may submit timestamped logs of auxiliary events, like, for example, the resizing of a window or the pausing of a video. These logs may be used to collect annotations at non-regular intervals or to collect analytics with the purpose of improving the Covfee task. Non-continuous task responses may also be recorded.

Covfee's key manager makes it easy to attach event handlers to keyboard and gamepad key presses. This is specially important for continuous annotation tasks, many of which must react to button presses.

Access to Covfee's admin panel which allows to keep track of progress and download annotation results and HIT URLs easily.

Reusability Covfee tasks are modular and configurable via the JSON specification and could be incorporated as part of Covfee to be reused by others.

Covfee's socket.io module allows the implementation of multiparty tasks, where multiple subjects take part in a task at the same time. The main use case for multiparty features is not annotation but the recording of live online interactions (written, audio or audiovisual) with the ability to query subjects at any point or request their live feedback.

4. Case studies

To illustrate the potential of the Covfee framework we present two case studies showcasing two custom annotation techniques: keypoint annotation and social action and confidence ratings.

4.1. Case study I: keypoint annotation in group interaction settings

In this case study, we focus on the task of labelling body joints or skeleton keypoints, particularly in the context of social interaction settings where precise, smooth annotation of keypoints over time is crucial. Manual keypoint annotations are particularly useful in the labelling of dense crowded scenes observed from the top-down view where interpersonal occlusion is minimised at the expense of more self-occlusion and more extreme perspective distortion effects.

Due to the bad performance of pose estimation methods in top-down videos, automatic extraction of body keypoints is often not an option in social interaction datasets.

To implement keypoint annotation continuously, the first challenge is the difficulty of following body joints in real time, with a mouse or other signaling device. Different body parts have different motion characteristics.

For example, hands and upper-body joints are used for gesturing, which can be characterized by sudden changes in velocity and acceleration, while shoulders exhibit smoother movements, and feet can be static for long periods of time when subjects stand still. Being able to annotate all of these accurately is vital for characterizing body movements in relation to speech. While annotating the video in slow motion would likely improve accuracy, we would like to avoid making the annotation process significantly longer. An ideal case would be if the video could be slowed down or sped up dynamically according to the speed of the keypoint that is being annotated. While this could be considered rather a chicken and egg problem since we do not yet know the speed of the object we are intending to track, we propose a method below which provides a solution to this problem.

4.1.1. Method

Covfee solves the problem of continuously annotating keypoints via a new annotation technique, which involves automatically adjusting the playback rate of the video in real time, according to the magnitude of the optical flow around the mouse cursor. We thereby leverage the fact that the annotator will be pointing the cursor at the keypoint of interest and use optical flow magnitude as an approximation to the speed of the target keypoint. The video playback rate is adjusted such that it is higher when the optical flow is high and lower when optical flow is low around the mouse cursor. This has the effect of slowing down the video when the joint being tracked moves fast and speeding up the video for slow-moving or static joints. It allows the users to annotate slow-moving joints at multiples of real time rate (eg. 4x playback rate), and fast gestures at fractions of it (eg. 0.1x playback rate) on the fly without additional user intervention.

Concretely, for a cursor position x, y (in pixels) at frame f, an NxN neighborhood in the vicinity of (x, y) is considered such that the playback rate at frame $f + 1$ is given by:

$$\hat{r}_{f+1} = C \sum_{i=x-N/2}^{x+N/2} \sum_{j=y-N/2}^{y+N/2} |O_{f,i,j}|$$

where $O_{f,x,y}$ is the optical flow vector for frame f at image location (x, y) and C is a constant. The best value of N depends on the video being annotated, and is a configurable parameter.

This rate is additionally bounded to prevent extremely low or high playback rates and a user-controllable multiplier C_u is added to allow the user to control the overall playback rate:

$$r_f = C_u \; max(r_{min}, min(r_{max}, \hat{r}_f))$$

This can only be implemented efficiently in an online setting if the flow computation is done offline and only the local averaging is calculated in the user's machine. For this, Covfee makes use of a pre-computed optical flow video, which is processed in the browser making use of a Javascript version of OpenCV.js (Bradski, 2000).

4.1.2. Study

This study presents results from applying continuous keypoint annotation, implemented in Covfee, to the annotation of keypoints in a human interaction dataset recorded during a professional social networking event. We start by comparing Covfee to a traditional, non-continuous approach using the CVAT tool on a small subset of the dataset, with annotation time and agreement as main variables of interest. Second, we analyze the application of Covfee to the complete dataset.

The dataset used, among other modalities, contains top-down video recordings from 48 subjects, interacting freely at the same time, as shown in Figure 1. The interaction space was recorded by 8 cameras for 45 minutes.

Our comparison consisted in the annotation of body joints for two data subjects in the same 20s video by two sets of three annotators: one set used CVAT, the other used Covfee. Annotators who used the continuous method were recruited from the Prolific crowd-sourcing

platform, without any filtering, and provided with a link to a HIT in Covfee. Because the CVAT tool does not implement support for crowd-sourcing, annotations in the CVAT condition were performed locally by three of the authors. No annotators had previous experience with any of the tools and work conditions were not controlled. Although crowd-sourced workers may have had previous experience in other kinds of annotation, we think the difference between our continuous keypoint annotation task and most crowd-sourced tasks is significant enough to make this experience unlikely to be a source of bias.

All annotators were provided written instructions to label the left shoulder, right shoulder, center of the head and a point in the direction of the gaze of the data subject (ie. in the direction of the nose). The goal with this last point was not to measure its precise location in pixels, but to use it to obtain a head orientation vector. Local annotators were asked to measure their total annotation time for CVAT. For Covfee, the time was acquired from the difference between the timestamps that Covfee adds to each data point.

In the case of CVAT, frames were annotated every second and linearly-interpolated in between. For Covfee, the method in Section 4.1.1 was used with parameters $N = 20$ (pixels), $r_{min} = 0.1$, $r_{max} = 4$. The video was pre-processed by denoising with the *hqdn3d* filter in FFMPEG (FFmpeg, 2016) with a temporal luma strength $luma\_tmp = 30$.

The annotators reported lower annotation times on average for the continuous approach (7.4min) compared to taking between 17 and 25 minutes for the CVAT annotations. We compared the annotations for head and shoulder key points by computing the average Euclidean distance in pixels between time-corresponding annotations. We averaged this discrepancy for all pairs of annotators. On average, our continuous annotation approach resulted in lower discrepancy (18.7 ± 10.0) when compared to the use of CVAT (22.9 ± 12.7), although within standard deviation.

The same was true when we measured discrepancy in the orientation of body and head in degrees (7.9 ± 4.8 for Covfee vs 9.9 ± 10.7 for CVAT). Body orientation was computed by taking the vector between both shoulder points, and head orientation was computed by taking the vector between the head keypoints and the gaze direction keypoint. Table 1 shows the errors measured per keypoint and annotation times in Covfee and CVAT. Annotation times for CVAT were not measured per keypoint. However, given that the CVAT annotations were image-based with a fixed interval between images, we expect annotation times to be roughly equal across keypoints (5.25min on average). It is particularly noteworthy that the head keypoint took on average significantly longer to annotate using Covfee, which is likely to be due to the head moving more rapidly during these segments. Even though our annotator sample was too small to measure differences in discrepancy across conditions, we are confident that the large (significant) differences in annotation time generalize to other situations. Even if true annotation quality were to be lower for the continuous case, we think the gains in annotation time are enough to make this an attractive approach for large scale annotation.

Given the results of the previous comparison, we proceeded to use Covfee to annotate body joints in the complete dataset. A total of 17 body keypoints (joints) were annotated for each subject, for a subset of 16 minutes of the dataset. This was equivalent to more than 218 hours of single-keypoint tracks.

Having an *occlusion* signal for each keypoint is important in the training of pose estimation methods (see Section 2.1). To support this important signal, we integrated a

Body joint	CVAT disc.	Covfee disc.	Covfee time	CVAT time
Head (px)	12.6 (7.9)	14.4 (12.0)	4.2min	5.25min
Left shoulder (px)	21.4 (11.1)	19.7 (6.9)	1.5min	5.25min
Right shoulder (px)	34.5 (19.1)	22.1 (11.2)	1.7min	5.25min
Head orientation (deg)	11.4 (12.8)	7.3 (4.0)	N/A	N/A
Body orientation (deg)	8.3 (19.9)	8.4 (5.6)	N/A	N/A

Table 1: Results of the comparison between Covfee continuous annotation and CVAT in the annotation of body joints. Values in parenthesis are standard deviations. Annotation discrepancies are averaged distances between corresponding annotations, over all pairs of annotators. Lower discrepancies indicate higher agreement. CVAT times are averaged since only totals (for all four annotated keypoints) were reported and we expect annotation times to be roughly constant for the four body joints. Note that the head and body orientation are derived values, hence no time is reported.

binary occlusion label into our technique by recording an additional key press. We asked annotators to hold down a keyboard key (while following the keypoint with their cursor) when the target joint was occluded. If the joint was still within the frame despite being occluded, annotators were asked to follow it approximately by inferring its location. While these annotations would in principle be *filtered out* of the training process, asking annotators to infer location in this way enables them to maintain continuity of the annotation. Additionally, though not standard practice, pose estimation methods could be trained to estimate occluded keypoints in addition to visible ones.

Adding this additional input made the annotation process slightly more involved, although in our pilot tests we did not notice any cognitive load issues with simultaneously following a keypoint with the mouse and annotating occlusion with the keyboard. Figure 4 shows the mean occlusion levels annotated over the image plane, averaged over our multiple videos. These plots use the same color scale and show the spatial variation in occlusion levels for body keypoints: head and feet. Continuous occlusion annotations allowed us to obtain a richer description of the skeletal data without increasing annotation time.

In summary, this case study of keypoint annotations showcases how Covfee is able to support a continuous annotation procedure that provides richer and better quality information about human body movements during socializing. This is in part due to the time efficient nature of the continuous annotation process, which allows for additional annotations to be made that can help us to understand and characterise better the relationship between the phenomena that are being labelled and the annotation noise.

4.2. Case study II: Social Action Annotation

The annotation of speaking status is particularly key in automated social interaction inference tasks. However, recording audio of people in real life settings can be very privacy invasive. Fortunately, from past efforts (Cabrera-Quiros et al., 2018) we know that it is possible to annotate speaking status from video only with some degree of annotator agree-

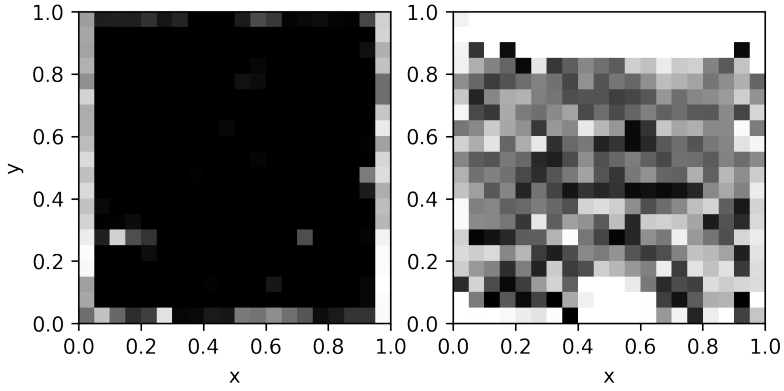

Figure 4: Plot showing the distribution of our occlusion annotations for head (left) and feet (right) keypoints. White indicates high occlusion and black low occlusion values. The head keypoint, being visible from most angles shows little occlusion while the feet tend to be more occluded when near the edges of the frame and show overall higher occlusion values.

ment sufficient for training machine perception systems (Gedik et al., 2019), although short back channels can be difficult to capture (Cabrera-Quiros et al., 2018). Acceptable inter-annotator agreement from video only can be explained by the fact that when humans speak, their vocal behaviour is often accompanied by linguistically related body movements such as gestures (McNeill, 1994).

This Section describes a case study about the annotation speaking status from video in a large social interaction dataset. The action of speaking was annotated using binary continuous annotation, where annotators were asked to hold down a keyboard key whenever they perceived speaking to be happening in the video.

In real life in-the-wild settings, videos may not always capture the subject of interest very clearly. The person may be partially occluded by others in the scene, they may have their back to the camera, or their face may not be visible. Access to multiple viewpoints of the data subjects is desirable to offset these challenges. This is however not a complete solution as in some cases none of the views may offer a suitable view of the subject of their speaking behavior could be hard to discriminate. This is a common situation with data recorded in real-life settings when intrusive sensing is avoided to preserve the naturalness of the interactions. To capture this uncertainty it would be of great benefit to know the confidence of the annotator in their judgement. To this end, we obtained continuous confidence ratings by asking annotators to indicate the degree of confidence that their action assessment (either speaking or non-speaking) was correct. In a training stage, such a confidence signal can be used to give less weight to data samples or segments for which the annotator had low confidence on being correct.

4.2.1. Method

Covfee supports action annotation via an interface for binary continuous annotation. The annotator is able to control the binary status of the annotation via a keyboard key: *true* if the key is pressed; *false* if it is not. Visual feedback is provided when the key is pressed.

Confidence annotations were also performed continuously in Covfee using an interface designed in general for continuous-value annotations. In this interface the users are able to control a vertical slider using their mouse. The vertical position of the slider follows the cursor's vertical position. The continuous value of the slider indicator (in the range $[0, 1]$) was recorded in Covfee.

4.2.2. Study

Our study on actions is based on the data obtained from the annotation of a large dataset (see Section 4.1.2) for speaking. Annotators were part of a larger group who worked on the annotation of our dataset, both for keypoints and speaking status. We selected conscientious annotators for this group via a short qualification task consisting on keypoint annotation only, and revised manually via playback of their annotations, but otherwise no special selection of annotators was done, nor did we control their working conditions. Annotators from the larger group worked on action annotation based on their availability when this phase of the project was reached.

In the action annotation stage, annotators were instructed to annotate the speaking status of all subjects in the scene, and to continuously *annotate* their confidence in their judgement about speaking status, per the method described above. To offset the issue of lack of visibility of the target subject, we gave annotators access to several side-elevated views of the subjects, from which they could pick the best one.

Computing turn lengths from the obtained speaking status annotations revealed that a high proportion of turn lengths were below one second in length, suggesting that we were able to capture quick turns, and potentially back-channels. Although we do not have access to speaking ground truth to verify it, our confidence annotations give us annotator ratings of their degree of certainty in their inferences. Figure 5 plots the turn lengths obtained from our annotations against the average confidence annotated (by the same annotator) during the corresponding turn. The plot does not reveal a clear trend, suggesting that confidence was not heavily dependent on turn length. It is likely that other factors like visibility may influence annotator confidence more.

5. Conclusion and Discussion

In this paper we have presented Covfee, a new web-based framework with the goal of supporting the study and use of continuous annotation in human behavior data. Although continuous-time annotation has long been used for affective dimensions, we present Covfee as a general framework, capable of supporting both these established techniques and new continuous annotation techniques. The motivation to support novel continuous annotation techniques for human behavior datasets (eg. for body joint and action annotation) comes from the potential to improve the time-efficiency of the annotation process when single

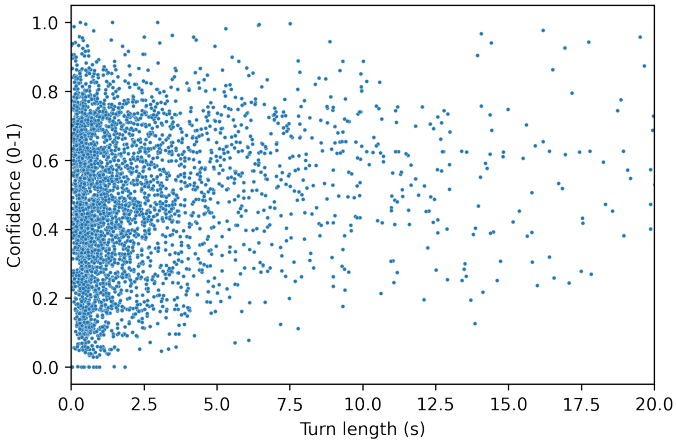

Figure 5: Plot showing the correlations between annotated turn lengths and mean annotation confidence during the turn for our speaking status annotations.

subjects are annotated for long periods of time (minutes to hours); and the suitability of continuous techniques for annotations that rely heavily on temporal context.

We have laid out the design decisions and main features of our framework, aimed both at basic users without knowledge of web development who wish to use existing tasks out of the box, or those with web development skills who wish to build new annotation techniques on top of Covfee. We started by explaining the workflow for requesters to use Covfee, which revolves around a specification file describing the HITs to be created. Covfee processes the specification file to create the annotation interfaces specified in it, and makes HITs available under a secure URL. We go on to explain how the tool supports data privacy and security for the annotation of potentially personal or sensitive data. We emphasize the design choices and features that make Covfee suitable for a crowd-sourcing setting, and lay out the features that make Covfee a framework, applicable to the implementation of new continuous tasks.

We presented two case studies applying continuous annotation (and Covfee) to keypoint and action annotations in a social interaction dataset. Our study on keypoint annotation showed an improvement in annotation time without a significant difference in annotation quality. Furthermore, our continuous technique allowed us to annotate keypoint occlusion in the same pass. This auxiliary signal is very relevant for the pose estimation task, since methods are usually fed only the set of visible keypoints and the occlusion signal is used to filter the input to the method. In traditional techniques, occlusion annotations are limited in time resolution by the frequency at which frames are annotated and cannot be interpolated between annotated keyframes like continuous values can. Our continuous technique, in contrast, provides a higher-resolution signal indicating when each keypoint becomes visible or occluded.

Similarly, in our action annotation study we obtained continuous-valued annotation confidence labels together with our binary speaking status annotations, although this time in a second pass over the data. Confidence signals give the researcher access to a measure of

uncertainty in the data labeling at each moment in time, without having to label the data multiple times to obtain an agreement based measure. One important question for future work is how well ratings of confidence from a single annotator approximate agreement measured from multiple annotators. Continuous-value annotations are also known to be affected by bias when interpreted absolutely (see Section 2.3). The extent to which this bias affects ratings of confidence including ours, and the best way to elicit and interpret confidence annotations are also open questions.

Although our study on actions did not involve an annotation time comparison with traditional action annotation techniques, we think that continuous annotation may also be a more time-efficient way to annotate most human actions since it can be done in real time, or even fast motion without the need to pause for labeling. Importantly, we think that time-efficiency should not be the only consideration when deciding for or against a continuous technique. In our experience, the suitability of continuous annotation for actions depends on the desired precision, frequency and context-dependency of the actions being annotated.

Regarding temporal precision, this is usually a function of the research questions being investigated. In human behavior research certain research questions involve the precise localization of action onsets and offsets, where onset and offset are reasonably well-defined and observable. Studies on the internal structure of gestures and laughter episodes, for example, make use of fine-grained temporal segmentation (Truong et al., 2019). In this case continuous annotation alone might not be a suitable solution given the annotation delays involved. Continuous annotation may however still be useful when the annotation task can be separated into two steps; first a continuous localization step (where actions are localized roughly in time) followed by a second precise temporal segmentation step, where a precise coding scheme is applied. In other words, at present we do not envision continuous action annotation as a complete solution for behavioral coding, but rather as a method for rough time-localization of phenomena of interest. In many machine learning applications, however, precise localization of action boundaries and action segmentation is not a requirement and robust machine learning methods or correction techniques have been proposed to mitigate the effects of delay in continuous annotations (Mariooryad and Busso, 2015; Khorram et al., 2019).

Regarding action frequency, continuous annotation provides greater time improvements the more frequent the target actions are. For extremely sparsely-occurring actions the time gain from continuous annotation becomes lower, as even in the non-continuous case, annotators would spend most of the time watching the media, and less time annotating. However, many actions of interest in human behavior research are frequent enough to benefit greatly from continuous annotation in terms of time-efficiency. In social signal processing and affective computing, actions such as speaking, gesturing, laughing, and other common actions in a social context are often annotation targets.

Finally, with respect to the temporal-context-dependency of the actions, we think continuous approaches are advantageous for most actions occurring in a social context because they enable the annotator to follow the flow of what is occurring in the interaction without interruption. Annotation of actions or situations such as "use of humour" or "enjoyment" requires a complex context-based judgement on the part of the annotators. Such context-heavy constructs are however common annotation targets in communities working with in-the-wild data such as social signal processing or affective computing.

Given these trade-offs we argue that continuous annotation is much more useful for action annotation than it's current usage would suggest.

It is important to highlight once again, however, that continuous annotation may not be suitable for every problem. The standard technique of bounding box annotation, for example, does not straight-forwardly translate to the continuous case since it is not clear how an annotator would control the location and dimensions of the bounding box continuously. This task is also hard to decompose into single-point annotation tasks since the corners of the box may not correspond to any meaningful keypoints in the scene. We cannot rule out, however, that new creative techniques will make it possible to perform such annotations continuously. Hybrid techniques where manual annotation is aided by models are not new and the application of such approaches to continuous annotation may open the door to new breakthroughs in annotation efficiency.

In general, Covfee has the long-term goal of dramatically improving the time and effort necessary to collect and annotate human behavior data online. It was born out of the need for a web annotation platform flexible enough to accommodate the high diversity and specificity of annotation needs present today. We expect that all of the design decisions made to support this goal will enable the adoption of Covfee as a platform for a) the implementation of existing annotation techniques such as those traditionally used within the affective computing community, b) experimentation with novel annotation techniques for vision tasks, such as the two techniques presented in this paper and c) developments in other fields such as the annotation of audio or other time series.

Acknowledgments

We thank Dr. Ekin Gedik for his role in setting up the CVAT server used for comparison in section 4.1.2. We thank Dr. Bernd Dudzik and Tiffany Matej for testing and providing feedback on preliminary versions of the annotation tasks in Covfee. This research is supported by the Netherlands Organization for Scientific Research (NWO) under project number 639.022.606.

References

Mojtaba Khomami Abadi, Ramanathan Subramanian, Seyed Mostafa Kia, Paolo Avesani, Ioannis Patras, and Nicu Sebe. DECAF: MEG-Based Multimodal Database for Decoding Affective Physiological Responses. *IEEE Transactions on Affective Computing*, 6(3):209–222, July 2015. ISSN 1949-3045. doi: 10.1109/TAFFC.2015.2392932. Conference Name: IEEE Transactions on Affective Computing.

Xavier Alameda-Pineda, Jacopo Staiano, Ramanathan Subramanian, Ligia Batrinca, Elisa Ricci, Bruno Lepri, Oswald Lanz, and Nicu Sebe. SALSA: A novel dataset for multimodal group behavior analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1707–1720, 2016. ISSN 01628828. doi: 10.1109/TPAMI.2015.2496269. arXiv: 1506.06882 ISBN: 1939-3539 (Electronic) 0098-5589 (Linking).

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. *Proceedings of the IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition, pages 3686–3693, 2014. ISSN 10636919. doi: 10.1109/CVPR.2014.471. ISBN: 9781479951178.

Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 279–283, Tokyo Japan, October 2016. ACM. ISBN 978-1-4503-4556-9. doi: 10.1145/2993148.2993165. URL https://dl.acm.org/doi/10.1145/2993148.2993165.

Brandon M. Booth and Shrikanth S. Narayanan. Fifty Shades of Green: Towards a Robust Measure of Inter-annotator Agreement for Continuous Signals. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 204–212, Virtual Event Netherlands, October 2020. ACM. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3418860. URL https://dl.acm.org/doi/10.1145/3382507.3418860.

G. Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000. tex.citeulike-article-id: 2236121 tex.posted-at: 2008-01-15 19:21:54 tex.priority: 4.

Alec Burmania, Srinivas Parthasarathy, and Carlos Busso. Increasing the Reliability of Crowdsourcing Evaluations Using Online Quality Assessment. *IEEE Transactions on Affective Computing*, 7(4):374–388, 2016. ISSN 19493045. doi: 10.1109/TAFFC.2015.2493525. Publisher: IEEE.

Carlos Busso, Srinivas Parthasarathy, Alec Burmania, Mohammed AbdelWahab, Najmeh Sadoughi, and Emily Mower Provost. MSP-IMPROV: An Acted Corpus of Dyadic Interactions to Study Emotion Perception. *IEEE Transactions on Affective Computing*, 8(1):67–80, January 2017. ISSN 1949-3045. doi: 10.1109/TAFFC.2016.2515617. Conference Name: IEEE Transactions on Affective Computing.

Laura Cabrera-Quiros, Andrew Demetriou, Ekin Gedik, Leander van der Meij, and Hayley Hung. The MatchNMingle dataset: a novel multi-sensor resource for the analysis of social interactions and group dynamics in-the-wild during free-standing conversations and speed dates. *IEEE Transactions on Affective Computing*, pages 1–17, 2018. ISSN 19493045. doi: 10.1109/TAFFC.2018.2848914.

Zhe Cao, Tomas Simon, Shih En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua(Xxx):1302–1310, 2017. doi: 10.1109/CVPR.2017.143. arXiv: 1812.08008 ISBN: 9781538604571.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The AMI Meeting Corpus: A Pre-announcement Machine Learning for Multimodal Interaction. *Machine Learning for Multimodal Interaction SE - Lecture Notes in Computer Science*, 3869:28–39, 2006. doi: doi:10.1007/11677482.3. URL citeulike-article-id:6473361%5Cnhttp://dx.doi.org/10.1007/11677482_3. ISBN: 978-3-540-32549-9.

Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.502. arXiv: 1705.07750 ISSN: 0032082X.

Yu Cheng, Bo Yang, Bo Wang, Yan Wending, and Robby Tan. Occlusion-Aware Networks for 3D Human Pose Estimation in Video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 723–732, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00081. URL https://ieeexplore.ieee.org/document/9010921/.

Roddy Cowie, Ellen Douglas-Cowie, Susie Savvidou, Edelle Mcmahon, Martin Sawey, and Marc Schröder. "Feeltrace": An instrument for recording perceived emotion in real time. *ISCA Workshop on Speech & Emotion*, (January 2000):19–24, 2000. doi: citeulike-article-id:3721917.

Roddy Cowie, Martin Sawey, Cian Doherty, Javier Jaimovich, Cavan Fyans, and Paul Stapleton. Gtrace: General Trace Program Compatible with EmotionML. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 709–710, September 2013. doi: 10.1109/ACII.2013.126. ISSN: 2156-8111.

Hao Shu Fang, Shuqin Xie, Yu Wing Tai, and Cewu Lu. RMPE: Regional Multi-person Pose Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2353–2362, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.256. arXiv: 1612.00137 ISBN: 9781538610329.

FFmpeg. ffmpeg tool, 2016. URL http://ffmpeg.org/.

Max Planck Institute for Psycholinguistics. ELAN [Computer software]., 2021. URL https://archive.mpi.nl/tla/elan.

Ekin Gedik, Laura Cabrera-Quiros, and Hayley Hung. No-Audio Multimodal Speech Detection task at MediaEval 2019, 2019.

Jeffrey M Girard. CARMA: Software for continuous affect rating and media annotation Jeffrey. 2(1):1–11, 2018. doi: 10.5334/jors.ar.CARMA.

Jeffrey M. Girard and Aidan G. Aidan. DARMA: Software for dual axis rating and media annotation. *Behavior Research Methods*, 50(3):902–909, 2018. ISSN 15543528. doi: 10.3758/s13428-017-0915-5. Publisher: Behavior Research Methods ISBN: 1342801709.

Pranay Gupta, Anirudh Thatipelli, Aditya Aggarwal, Shubh Maheshwari, Neel Trivedi, Sourav Das, and Ravi Kiran Sarvadevabhatla. Quo Vadis, Skeleton Action Recognition ? *arXiv:2007.02072 [cs]*, July 2020. URL http://arxiv.org/abs/2007.02072. arXiv: 2007.02072 version: 1.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense Human Pose Estimation in the Wild. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. ISSN 10636919. doi: 10.1109/CVPR.2018.00762. arXiv: 1802.00434 ISBN: 9781538664209.

Zhaocheng Huang, Ting Dang, Nicholas Cummins, Brian Stasak, Phu Le, Vidhyasaharan Sethu, and Julien Epps. An Investigation of Annotation Delay Compensation and Output-Associative Fusion for Multimodal Continuous Emotion Prediction. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 41–48, Brisbane Australia, October 2015. ACM. ISBN 978-1-4503-3743-4. doi: 10.1145/2808196.2811640. URL https://dl.acm.org/doi/10.1145/2808196.2811640.

Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic Studio: A Massively Multiview System for Social Interaction Capture. *arXiv:1612.03153 [cs]*, December 2016. URL http://arxiv.org/abs/1612.03153. arXiv: 1612.03153.

json-schema org. JSON Schema, 2020. URL https://json-schema.org/.

Soheil Khorram, Melvin McInnis, and Emily Mower Provost. Jointly Aligning and Predicting Continuous Emotion Annotations. *IEEE Transactions on Affective Computing*, 3045 (c):1–16, 2019. ISSN 19493045. doi: 10.1109/TAFFC.2019.2917047. arXiv: 1907.03050 Publisher: IEEE.

Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, January 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.15. Conference Name: IEEE Transactions on Affective Computing.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*, February 2015. URL http://arxiv.org/abs/1405.0312. arXiv: 1405.0312.

Phil Lopes, Georgios N. Yannakakis, and Antonios Liapis. RankTrace: Relative and unbounded affect annotation. *2017 7th International Conference on Affective Computing and Intelligent Interaction, ACII 2017*, 2018-Janua:158–163, 2018. doi: 10.1109/ACII.2017.8273594. ISBN: 9781538605639.

Reza Lotfian and Carlos Busso. Building Naturalistic Emotionally Balanced Speech Corpus by Retrieving Emotional Speech from Existing Podcast Recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483, October 2019. ISSN 1949-3045. doi: 10.1109/TAFFC.2017.2736999. Conference Name: IEEE Transactions on Affective Computing.

Soroosh Mariooryad and Carlos Busso. Correcting time-continuous emotional labels by modeling the reaction lag of evaluators. *IEEE Transactions on Affective Computing*, 6 (2):97–108, 2015. ISSN 19493045. doi: 10.1109/TAFFC.2014.2334294. Publisher: IEEE.

Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schröder. The SEMAINE database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1): 5–17, 2012. ISSN 19493045. doi: 10.1109/T-AFFC.2011.20. ISBN: 1949-3045.

Gary McKeown, William Curran, Johannes Wagner, Florian Lingenfelser, and Elisabeth Andre. The Belfast storytelling database: A spontaneous social interaction database with laughter focused annotation. *2015 International Conference on Affective Computing and Intelligent Interaction, ACII 2015*, (January 2016):166–172, 2015. doi: 10.1109/ACII.2015.7344567. ISBN: 9781479999538.

David McNeill. Hand and Mind: What Gestures Reveal About Thought. *University of Chicago Press*, 1994. ISSN 00238309. doi: 10.1177/002383099403700208.

David Melhart, Antonios Liapis, and Georgios N. Yannakakis. PAGAN : Video Affect Annotation Made Easy. 2019. arXiv: 1907.01008v1.

Angeliki Metallinou and Shrikanth Narayanan. Annotation and processing of continuous emotional attributes: Challenges and opportunities. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013*, (EmoSPACE), 2013. doi: 10.1109/FG.2013.6553804. ISBN: 9781467355452.

Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160, June 2011. doi: 10.1109/CVPR.2011.5995586. ISSN: 1063-6919.

Jonathan Posner, James A. Russell, and Bradley S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17(3):715–734, 2005. ISSN 0954-5794. doi: 10.1017/S0954579405050340. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2367156/.

Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013*, (i):1–8, 2013. ISSN 2326-5396. doi: 10.1109/FG.2013.6553805. Publisher: IEEE ISBN: 9781467355452.

Fabien Ringeval, Björn Schuller, Michel Valstar, Jonathan Gratch, Roddy Cowie, Stefan Scherer, Sharon Mozgai, Nicholas Cummins, Maximilian Schmitt, and Maja Pantic. AVEC 2017: Real-life Depression, and Affect Recognition Workshop and Challenge. In *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*, AVEC '17, pages 3–9, New York, NY, USA, October 2017. Association for Computing Machinery. ISBN 978-1-4503-5502-5. doi: 10.1145/3133944.3133953. URL http://doi.org/10.1145/3133944.3133953.

James A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980. Publisher: American Psychological Association.

Karan Sharma, Claudio Castellini, Egon L. van den Broek, Alin Albu-Schaeffer, and Friedhelm Schwenker. A dataset of continuous affect annotations and physiological signals for emotion analysis. *Scientific data*, 6(1):196, 2019. ISSN 20524463. doi: 10.1038/s41597-019-0209-0. arXiv: 1812.02782.

Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. *arXiv:1604.01753 [cs]*, July 2016. URL http://arxiv.org/abs/1604.01753. arXiv: 1604.01753.

Khiet P Truong, Jurgen Trouvain, and Michel-pierre Jansen. Towards an annotation scheme for complex laughter in speech corpora. pages 529–533, 2019.

Jose Vargas Quiros. Covfee: Continuous Video Feedback Tool, 2021. URL https://josedvq.github.io/covfee.

Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently Scaling up Crowdsourced Video Annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013. ISSN 0920-5691. doi: 10.1007/s11263-012-0564-1.

Lu Zhou, Yingying Chen, Yunze Gao, Jinqiao Wang, and Hanqing Lu. Occlusion-Aware Siamese Network for Human Pose Estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12365, pages 396–412. Springer International Publishing, Cham, 2020. ISBN 978-3-030-58564-8 978-3-030-58565-5. doi: 10.1007/978-3-030-58565-5_24. URL https://link.springer.com/10.1007/978-3-030-58565-5_24. Series Title: Lecture Notes in Computer Science.