OULUN YLIOPISTO
UNIVERSITY of OULU

Online Annotations Tools for Micro-Level Human Behavior Labeling on Videos

University of Oulu
Faculty of Information Technology and
Electrical Engineering/M3S
Master's Thesis
Wenting Tao
04-08-2020

Abstract

Successful machine learning and computer vision approach generally require significant amounts of annotated data for learning. These methods including identification, retrieval, classification of events, and analysis of human behavior from a video. Micro-level human behavior analysis usually requires laborious efforts for obtaining the precise labels. As the quantity of online video grows, the crowdsourcing approach provides a method for workers without a professional background to complete the annotation task. These workers require training to understand implicit knowledge of human behavior. The motivation of this study was to enhance the interaction between annotation workers for training purposes. By observing experienced local researchers in Oulu, the key problem with annotation is the precision of the results. The goal of this study was to provide training tools for people to improve the label quality, it illustrates the importance of training. In this study, a new annotation tool was developed to test workers' performance in reviewing other annotations. This tool filters very noisy input by comment and vote feature. The result indicated that users were more likely to annotate micro behavior and time that refer to other opinions, and it was a more effective and reliable way to train. Besides, this study reported the development process with React and Firebase, it emphasized the use of more Web resources and tools to develop annotation tools.

Keywords
Annotation tool, human behavior, micro-level, crowdsourcing, React, Firebase.

Supervisor
Ph.D., University Lecturer, Raija Halonen

Foreword

I would like to thank my supervisor Dr. Raija Halonen for her supervision and support during my master's thesis study. Her insightful and valuable advice influenced my systematic thinking and writing. I would like to thank Dr. Maëlick Claes for his great help and valuable advice has deeply improvement of my master's thesis study. I would like to thank Dr. Antti Siirtola for encouraging me to insist on this topic. I would like to thank Pyry Pennanen for constructive discussions and help me with the development process. It was a challenge for me to learn this total new knowledge and skill. It was a meaningful experience for me.

Wenting Tao

Oulu, August 4th, 2020

Contents

1. Introduction

With Artificial Intelligence (AI) being embedded in our living and working spaces, the increasing capacity of machine learning and deep learning trends is to use enriched data to improve performance, such as deep neural networks, which usually require a large number of annotated video sequences to train the network parameters. Enriched data means collected with more features that can be integrated into a wide variety of devices. (Park & Yang, 2019.) Video becomes a popular source to provide information about a scene. Video with annotation is a very effective container of data, which is an important part of the AI foundation work, particularly in understanding human behavior. (Enser, 2000.) Successful machine learning along with computer vision approaches generally requires a significant amount of video annotated data to learn, including the identification, retrieval, and classification of events and the analysis of human behavior (Barnard, Duygulu, Forsyth, Freitas, & Blei, 2003).

Video annotation is a task that consisting of manually labeling videos, frame by frame or by short sequences, display the objects, object type, tracks of objects in the whole video. For example, it needs to examine what kind of the behavior of consumer refers to emotions, actions, movements, nonverbal vocalizations, and what are the people's current behaviors, how many people are involved, how they communicate with those around them, and what the environment affects them. (Pantic, Pentland, Nijholt, & Huang, 2007.) However, the labeling of human behaviors is challenging even for experts since the annotation tasks are extremely time-consuming. For example, a 5-minute video could take an hour for a single worker to properly annotate. Human gesture labels' segmentation and recognition are discrete, but the discrete labels still have to be related to the context of the whole video. Researchers expect to decrease the time that streamlines the process of the whole video and focus directly on the labels of interest. (Wang, Narayana, Smith, Draper, & Beveridge, 2018.)

One way to solve this problem is to use crowdsourcing, which is an online labor market that splits the project down into smaller tasks in the form of temporary work, tasks such as computing techniques, performance analysis, applications, algorithms, performance, and dataset. Employers set task prices, usually, the price is very low, and then post them for workers to browse and choose from. The general micro-level human behavior annotation method of crowdsourcing is to assign each task to different workers for independent annotation and then compare, check, summarize, and use specific methods for aggregation. (De Amorim, Segundo, Santos, & Tavares, 2017.) However, crowdsourcing faces a critical issue of how to control the annotation quality, as the annotation workers have different experiences and knowledge. Simple tasks can be accomplished by anyone in crowdsourcing, but the more complex ones still require trained workers, such as identifying and recognizing fuzzy micro-gestures and positioning them in the right position on a video, as well as complementary content. (Nowak & Ruger, 2010.)

Existing annotation tools all have something in common: the understanding of complex human behavior at the micro-level is difficult. Micro-level behaviors are the natural reaction of human emotions, which is hardly annotated with a fixed method of analysis. Especially some subtle non-verbal micro-expressions and micro postures are difficult to be judged. The level of training required for complicated human behavior annotation is high so that the non-professional user could not comprehend it. Unlike ordinary people, for the expert, the exact precise properties of the nature of the machine as well as the specific search method of knowledge are to be determined. (Park, Shoemark, & Morency, 2014.) Yet people are dependent on higher concepts. That is, the natural language is common to ordinary people. For non-professional workers, the use of natural language that free-form text to speculate human confused gesture and inconspicuous emotion is a facility task. Natural language labeling includes five aspects: recognition of results, word similarity, identification of text meanings, and time sequence. (Snow, O'connor, Jurafsky, & Ng, 2008.)

The previous annotation tools are still completing the work separately, they focus on using the deviation algorithm method to get the results close to the expert level. Discussing the accuracy of annotation puts too much emphasis on comparing the level of expertise. The statistical algorithms using mathematical tools do not solve and reduce the generation of labeling errors. (Park et al., 2014.) Previous tools have done little to support using natural language to descriptive annotation. Their feature limits communication and sharing. New worker learning is still a major issue. The general worker's perception of the annotation is not improved, since they do not share the annotation results. Annotation results are not used to the max. Interaction is required when users perform implicit and complex annotation tasks. (De Amorim et al., 2017.)

The annotation deviations of the different workers are inevitable, subjective factors like carelessness, spelling errors, and objective factors like interpreting complex micro gestures (Dasiopoulou, Giannakidou, Litos, Malasioti, & Kompatsiaris, 2011). This study's motivation was to provide a Web tool to present and discuss the annotation opinions of workers for knowledge sharing. Recent studies show that although using crowdsourcing to annotate complex human micro-level behavior with video tasks to reduces cost, manual work for ordinary people is inefficient (De Amorim et al., 2017).

Previous research suggests developing annotation tools requires the use of the Web's massive resources such as social media videos and corpora, and also use of the many web-based tool libraries services such as visual tools, statistical tools (Spiro, Taylor, Williams, & Bregler, 2010). The template annotation tasks need to be easy to build and change target tasks in different scenarios. The web annotation tool needs to using modular code development to be extensible and easily ports behavior annotations from videos to popular platforms. (Park, Mohammadi, Artstein, & Morency, 2012.)

The purpose of this study was to provide an interactive tool to support people without related industry background for micro-level human behavior labeling on video annotation training and learning. The research question of this study was:

How to develop a lightweight and interactive annotation tool for people to learn, comment, and vote about opinions on micro-level human behavior labeling?

This study presented a simplified annotation process to support workers with different backgrounds for training and learning. Through literature research, this study reported the need to help new users so that they can learn how to mark labels of high quality. The annotated results of trained workers are higher than the method of quality control. (Rashtchian, Young, Hodosh, & Hockenmaier, 2010.) Following the process of the design sciences research, a new annotation tool was developed and tested. The new annotation tool gathered the labels and opinions of the users through the comment feature. Crowd annotation has to face noisy and mistake input, This tool uses comment and vote function to filter very noisy input. (Park et al., 2014.)

This study results indicated that users were more likely to interact, and refer to other people's labels to annotate, and it was a more effective and reliable way to train. In social media, persuasion is at the core of interaction. Commenting and voting on the previous annotations is useful in a training system. Such systems can help more professional workers show more persuasively. Users can be trained to be better in annotating context and content. Analysis and feedback on existing annotations shown to users can have a significant impact on the availability and effectiveness. (Park et al., 2019.)

The contribution of this study including a new tool developed for the human micro gesture, micro expression, emotion, the precise start/end times annotation. Different previous tools highly rely on the annotation experts to control the quality (Park et al., 2014). In this study, comments and votes by different workers were emphasized for knowledge discussion improved the annotation quality. Moreover, this study addressed the semantic gap issue. The semantic gap is the distance between low-level and high-level requirements for retrieval, caused by the inconsistency between the computer's visual information of the image and the user's understanding of semantic information. (Smeulders, Worring, Santini, Gupta, & Jain, 2000.) People usually distinguish the similarity of images on the semantic understanding of the objects or events described. However, computers derive the visual features of images directly. This difference between humans and computers in understanding images generates a semantic gap. (Rashtchian et al., 2010.)

This thesis is organized into six parts. The background section is the literature review on human behavior understanding, labeling tools, crowdsourcing, semantic gap, and development tools: React and Firebase, one is used for developing a web app, the other as a database. The third section introduces the research method and reports the annotation tool design. In the fourth section, the study, implementation, and empirical research are analyzed via the development process that builds a fast responsive Web application, testing, and evaluation. Then, the results of the proposed method are discussed and concluded in the last section.

2. Background

This section presents an existing literature review related to this study, involving human behavior understanding, micro-level behavioral annotation, the existing annotation tools, crowdsourcing, semantic gap, and development tools. The reasons for errors in annotation results are analyzed and discussed. Furthermore, the problem of the semantic gap is highlighted. The use of crowdsourcing is the trend, but the quality of annotated results is low. (Park, Mohammadi, Artstein, & Morency, 2012.) Returning to manual annotation, developing an open annotation tool is urgent (Yuen, King, & Leung, 2011). These tools need to provide an interactive platform for learning, and training to users, and introduce the viewpoints of label results (Park et al., 2014). There are few efforts in previous annotation tools with consideration of these factors. The choice of development tools is also a core issue. (Dasiopoulou et al., 2011.) In this study, React and Firebase are selected for their superior network performance (React, 2020).

2.1 Artificial intelligence and annotations

With Artificial Intelligence (AI) being embedded in our living and working spaces, the critical realization of management services is to meet the customer needs, for example, how to make AI can predict the consumer's needs to improve service. If this prediction is to come true, then next-generation user interfaces will be human-centered. It will go beyond the keyboard and mouse to include natural, human-like interactive functions such as affective and social signals. (Pantic et al., 2007.) Once a new technology has evolved enough to fulfill its practical requirements, the next inevitable step is to make it more comfortable and efficient from a human perspective (Park, 2016).

It is assumed that the machine could understand consumer psychology and human behavior, the premise of this assumption is how to obtain a data source that has been accurately and reliably annotated (Pantic et al., 2007). Therefore, annotation research has been developed by the growing volume of annotated text corpora produced by projects and evaluation initiatives (Bontcheva, Cunningham, Roberts, Tablan, & Aswani 2013). Video annotation is a vital part of research in artificial intelligence, computer vision, machine learning, and interface design. Effective annotated data created by the user is increasingly being utilized for accurate content search and retrieval. (De Amorim et al., 2017.)

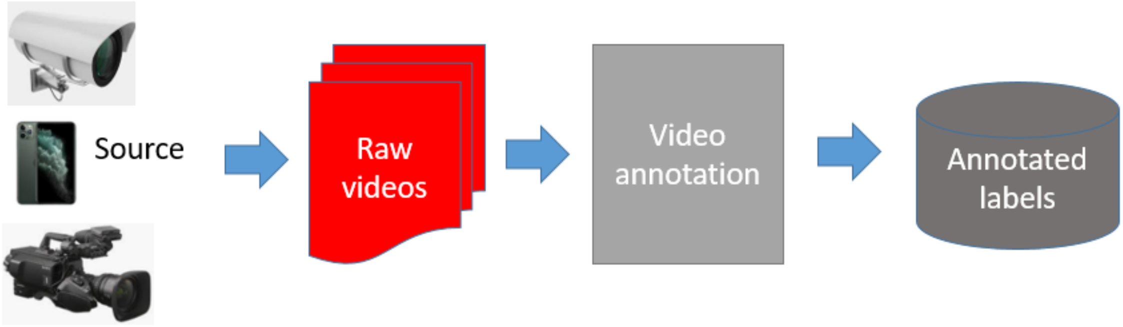


Figure 1. General video annotation.

Figure 1 is the general video annotation process, the original video resources are transformed into classified label data. With the promotion of Web social media, web-based video sharing platforms are very popular. Annotation workers can use the Web to obtain a large number of resources to add new tags and handle large-scale video annotations, exploiting the Web approach for mining the diversity of possible annotations. (Li et al., 2009.) The topic and content of each annotation can be in HTML format, so it can be multimedia or hyperlinked to other Web resources (Gao, Wang, Yong, & Gu, 2009). The key to the success of these applications is how to effectively and efficiently manage and store a huge amount of audio-visual information, while at the same time providing user-friendly access to the stored data (Li et al., 2009). The field of research known as video abstraction has rapidly emerged (Truong & Venkatesh, 2007). Although automatic video summarization techniques have been studied extensively, there is no lightweight method that can summarize any type of video clip effectively. Because it is extremely difficult to develop a program that can judge the semantic validity of the annotation results in general and set quality control. (Wu, Thawonmas, & Chen, 2011.)

2.2 Previous annotation tools

Different perspectives on semantic content and the variety of feature information result in various video annotation tools emerging. They are divided into scene, theme, event, action, prediction. The methods of labeling include the image, time, and object. In different fields, there are different annotation structures and functions to represent certain content according to specific requirements. The typical manual annotation tools focus on issues such as scope, information rate, granularity, vocabulary annotation, and annotation statements syntax. (Dasiopoulou et al., 2011.)

Mechanical Turk is an online labor market, where small sums of money are charged to workers to complete specific tasks. The system is as follows: one requires an Amazon account to send an annotation task or annotation submitted task. Amazon accounts are anonymous but have a unique Amazon ID. An applicant may generate a group of human intelligence tasks, each task consisting of an arbitrary number of issues. The client requests annotations for the task that determine the number of specific annotations per task that they are willing to pay for. (Snow et al., 2008.)

The VideoAnnex tool is under the respective MPEG-7 definition schemes. It supports descriptive, textual, and administrative annotations. Descriptive data refer to the entire video, different fragments of video, or regions within key-frames. (Lin, Tseng, & Smith, 2003.) It provides XML format default topic lexicons, allowing the user to build and load their XML lexicon, design a concept hierarchy via the commands of the interface menu, and insert free-text descriptions. It is interesting to note that this tool provides an extra feature, which is learning annotation. This functionality helps the annotation worker in identifying and labeling related shots with the same descriptions. (Dasiopoulou et al., 2011.)

The Ontolog tool is a video and audio source annotation system that includes standardized sets of terms or concepts. It is a Java application and explores different database types, including descriptive, structural, and administrative. (Heggland, 2002.) Illustrative annotations are added, imported, and generated by the user. The structure definition relating to the video clip is generated at a user-defined point, following a simplified structure. Ontolog introduced various search queries and retrieval processes. (Dasiopoulou et al., 2011.)

These existing tools cannot meet some crucial requirements for content annotation: for instance, the interoperability of the created data and the ability to automatically process them. The interoperability of the created data means the lack of interaction, including the capacity to share and reuse annotation or comment on other video annotations. The ability to automatically process determines the micro or macro level of expression content and the benefits produced from the available annotation. The ability to automatically process is key to the growth of smart content management services. Automatic processes are real-time segmentation and prediction, including semantic segmentation, automatic time interval segmentation, automatic object allocation, and tracking. The presence of commonly agreed vocabulary, grammar, terms, and structures are important elements for automatically process achievement. Though there are many forms of annotation tools, each tool selects a specific angle within a fixed field. The accuracy of the annotation result remains the ultimate challenge Precision is a prerequisite for the automated production of annotations. (Dasiopoulou et al., 2011.)

2.3 Micro-level behavioral annotation

The big problem with human behavior annotation is how to reasonably understand human behavior, the natural interactional function that is human-like. While there is agreement via various theories that at least some behavioral signals have evolved to communicate information, annotation is a difficult, time-consuming task that requires high cognitive effort. There is a lack of agreement regarding specificity micro-level behavior. Different people have different backgrounds in knowledge, the extent to which they are innate and universal, and whether they convey emotions, social motives, behavioral intentions, or all three. (Izard, 1997.) The highly debated issue is whether affective responses is a separate signal of behavioral communication like expressing feelings, or a related sign of behavior like facial expression (Fridlund, 1997).

Nonverbal behavior patterns are studied in terms of a feature in social interactions. Existing studies find that nonverbal behavior indicated an important manner for the expression of communicators' inner feelings and intentions. Nonverbal behavior can be defined as the affective reaction toward the other person in interpersonal communication. (Li, Lu, Zhang, Li, & Zhou, 2009.) Researchers argued that nonverbal signals are more believable than verbal cues as those are impulsive and harder to be manipulated (Cristani, Raghavendra, Del Bue, & Murino, 2013).

Explaining the internal state of the human behavioral signal, like an emotional state, is a standard flow of thought. Discrete emotion theorists suggest the existence of six or more basic emotions (happiness, anger, sadness, surprise, disgust, and fear) that have been universally expressed and recognized by non-verbal behavioral signals, especially facial and vocal expressions. (Juslin, Scherer, Harrigan, & Rosenthal, 2005.) Studying human communication is a large multidisciplinary activity with researchers from a wide range of backgrounds, including psychology, sociology, cognitive science, linguistics, and signal processing (Cristani et al., 2013). Modeling human behavior corpus sources is a difficult task (Rashtchian et al., 2010).

Typically, supervised learning methods include a great number of annotated video sequences. Although some of these algorithms are implemented at the scene level that is referred to as macro-level annotations, many of these problems need micro-level annotations to determine the precise start and end times of an event or behavior. So micro-level behavioral annotations are to identify the precise start, end time, and opinion of a behavioral cue in a given video sequence of human behavior. (Park et al., 2014.) Many labeling proposals are not linear and can be discussed separately (Lewis, Haviland-Jones, & Barrett, 2010). Compared to other video annotations, the micro-level annotations are more tedious and boring work, and the accuracy of annotations decreases with the difficulty of the process. Micro-level behaviors are the natural reaction of human emotions, which hardly is annotated with a fixed method of analysis. (Wang, Phan, Rahulamathavan, & Ling, 2017.) Even the correctness of professional annotation is different (Pantic et al., 2007).

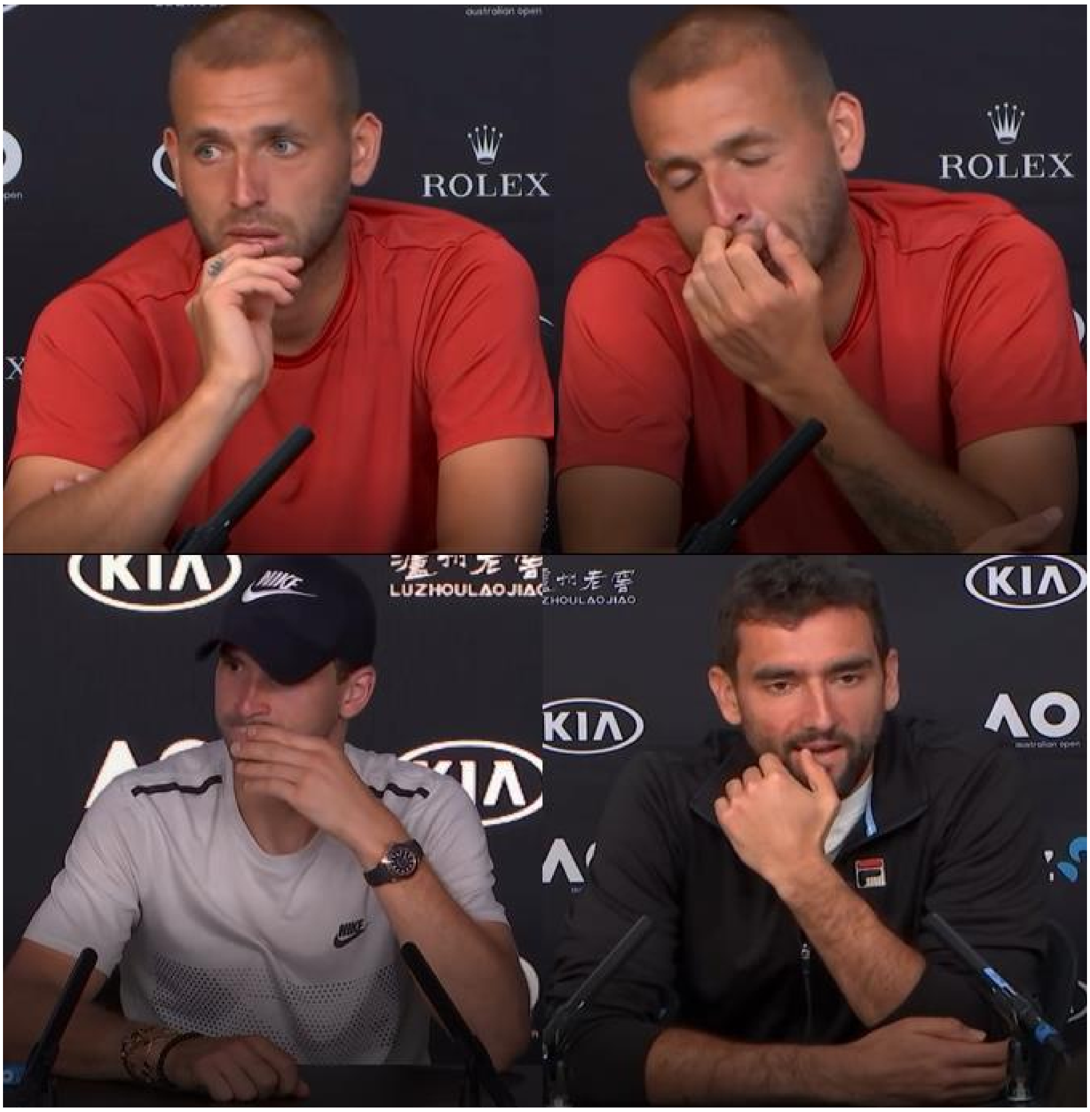


Figure 2. Confused micro-gestures which can be labeled as cover mouth, touching jaw, and biting nails.

Figure 2 is the four selected frames from videos to illustrate the confusion of micro-gestures. These gestures need to be identified as cover mouth, touching jaw, or biting nails. The annotation of human micro-level behavior is more complicated and problematic for ordinary or fresh workers since it is unlike standard semantics which is easy to capture, represent, and explain (Pantic et al., 2007). Non-verbal information is a means of analyzing and predicting human behavior, and previous research has found that people depend primarily on micro-expressions and micro postures to interpret the behavioral signals of a person, and are associated with social signals. Many human micro-expressions and gestures are unconscious and could be so subtle that they cannot be encoded or decoded. But human behavior must be a clear flow of information and commands to a computer. The semantic annotation of micro-level behavior is not only on syntax and linguistic interoperability but also on correct psychological behavior judgment. It shows that the interpretation of micro-level behavior in videos by common people mainly focuses on the level of semantic, ambiguity, fuzziness, and subjectivity that created obstacles to the analysis. (Wang et al., 2017.)

2.4 Crowdsourcing

Precise annotations are often made by experienced locals and are expensive on budget and time, since annotating videos manually is a time consuming and costly task, one way to solve this problem is to use crowdsourcing. Crowdsourcing platforms have proven to be a viable option when it came to providing access to a scalable workforce and ready annotation tools. (Vondrick, Patterson, & Ramanan, 2013.) Crowdsourcing is an online labor market that splits the project down into smaller tasks that are hard for computers to perform. Crowdsourcing practices are all done in the form of temporary work, tasks such as computing techniques, performance analysis, applications, algorithms, performance, and data sets can be divided into work. Employers make human intelligence tasks and set their prices, usually, the price is very low, and then post them for workers to browse and choose from. (Park et al., 2014.)

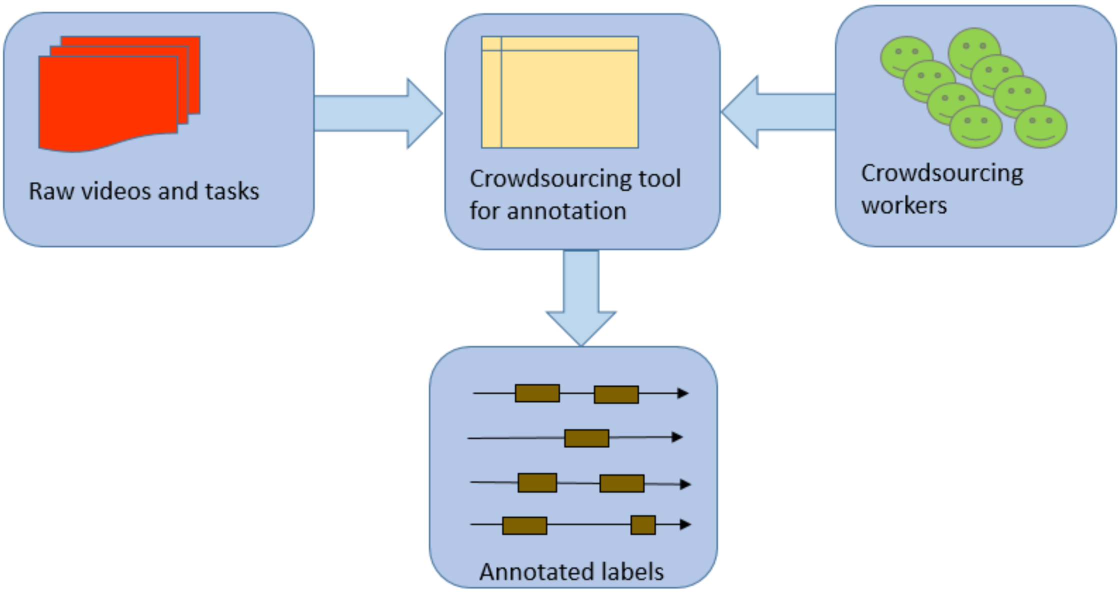


Figure 3. Approach for crowdsourcing annotations.

Figure 3 is an overview of the approach for crowdsourcing, the raw videos and tasks are assigned to people of different backgrounds, and the quality of the labels annotated are different. Crowdsourcing is an attractive solution to the problem of cheaply and quickly acquiring annotations to construct all kinds of predictive models. But if it ignores quality control, it may yield poor quality results, especially when crowdsourcing tasks are assigned to the non-professional worker. Since it is difficult and expensive to manually recheck the submitted results, distributing annotation to crowdsourcing platforms such as Amazon Machine Turk (AMT) takes quality at risk for requesters. (Park et al., 2012.) There is still no way to design how to produce better output for workers. Quality is a major problem with crowdsourcing, such as carelessness, deception, spelling mistakes, grammar mistakes, deviation of understanding. (Rashtchian et al., 2010.)

Previous studies have shown that financial incentives increase quantities but not quality. It has been shown that controls lead to lower communication (Rashtchian et al., 2010). Furthermore, existing technology cannot distinguish the true error rate (unrecoverable) from the error (recoverable) that some workers exhibit (Park et al., 2012). Besides, existing solutions such as AMT have no effective mechanism to guarantee workers such as proficient English writers. Therefore, it provides a limited way to avoid noisy comments. (Park et al., 2014.) A little research has been done to evaluate the agreement of micro-level annotations as well as the evaluation of annotation skills and reputation under the account of the client (Park et al., 2014). Since there is an adequate reorganization of the assets of the annotations and different approaches, the fusion of multiple clues like visual, audio clues, with other data for extracting the result of goal event, predicted target event represents a complete annotation result. If the final results of the produced annotations are aggregated, analysis is a necessary task that finds enough information from a huge label database. (Vondrick et al., 2013.) In the sense of monitoring a user's work, performance evaluation is very important as it is not easy to obtain shared videos and the corresponding reference information for tracking (Vezzani & Cucchiara, 2008).

2.5 Semantic gap

The image's low-level features refer to the outline and edges, computers derive the visual features of images directly. High-level semantics refers to transforming complete image content into a visually understandable expression of the text language. (Smeulders et al., 2000.) People usually distinguish the similarity of images on the semantic understanding of the objects or events described. This kind of understanding cannot be directly extracted from the visual features of the images. The semantic gap is because of the difference between humans and computers in understanding. (Rashtchian et al., 2010.)

The main reason for the differences in video stream interpretation is the knowledge and experience, as well as the results of low-level feature extraction. Consequently, many works of literature describe the use of diverse annotation methods to make up this semantic gap. (Rashtchian et al., 2010.) The semantic gap raises usability issues and whether the annotation results are effective. Since the disadvantage of video annotation is that strong bias, diversity of language and opinion will minimize bias and influence performance. (Snow et al., 2008.) Currently, the video annotation process becomes a very necessary task to overcome the problem of finding adequate information on a huge database. It is necessary to review the differences among users by showing the annotation process. (Rashtchian et al., 2010.) The differences are inevitable because the interpretation of the video stream is dependent on the awareness, experience, context, and other variations of the annotator, as well as the extraction of the key features. With the increase in annotation data size, the variety of record data, and the video sequence complexity, it is a challenge on how to create an efficient database. To achieve this, it is necessary to translate analysis results to a semantic meaning related to the video sequence. (Tani, Ghomari, & Tani, 2019.)

2.6 Development tool

Video annotation tools make very poor use of semantic Web technologies and formal context, a problem that affects not only the content concerned, but it also explains the advancement of semantic Web technology and new Web efficient services in the context of complex change in intelligent content management (Dasiopoulou et al., 2011). The reason for choosing React is that it has very strong community support. A large community of developers is making React better because it's an open-source library, and programmers from around the world are helping people learn the technology in different ways. (React, 2020.) At the time of writing, the repository had 1,207,501 contributed projects on GitHub (GitHub, 2020).

2.6.1 Web development tool: React

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. Composing complex UIs from small and isolated pieces of code called “components”. React has different kinds of components, it is designed with the concept of reusing components. Define small pieces and assemble them into large components. All components are reusable regardless of size, even across projects. If simplicity is the functional goal of this study tool, the framework structure could be simple and easy to implement. React has simple code logic and unique ideas. It is well suited to the lightweight and flexible requirements of this study, and it is easy to extend in the future. Successful companies like Facebook and PayPal use the React, which necessarily means it is a really popular library. (React, 2020.)

The componentization of React responds to the prevailing demand. React components are taken out of the DOM, even the SC (stateless component: the simplified component API is intended for components that are pure functions of their props.) and database become components that are treated as objects. As React is a class library, developers often need to leverage other class libraries to build a real application. React can combine short, discrete pieces of code into complex UI interfaces. (React, 2020.) These pieces of code are called ‘components’. Components can be seen as a function or an object, scoping it according to the single function principle. In other words, a component can only be responsible for one function in principle. (Gackenheimer, 2015.) For example, users use the reaction-router library to handle routing, import reaction-player library for video function. If users see it from a frame perspective, React is like buying a PC straight from the box or buying the parts to build it ourselves. The React only loads the parts users need. (React, 2020.)

Another distinct advantage of React is the front-end processing. React is a one-way data flow, which can be completed only by processing how to get the interface from the data. In this way, the interface and data can easily keep consistent, so the maintenance and management of component state can be clearer. (Gackenheimer, 2015.) One-way updates to annotated data should be easily controlled. Annotated data, comments, and vote updates in this study tool should render the update of the virtual DOM immediately. Data updates the DOM flow from the top level, so user events do not directly manipulate the DOM but manipulate the top-level data. The code rarely addresses DOM directly, but only with changes to the data. This greatly simplified the code and logic. For interactions, all need to do is update the data source. React passes the data down from the top component layer by layer. This mechanism is suitable for real-time display of video, annotations, comments, and voting updates, rather than dealing with complex logical operations. (React, 2020.)

2.6.2 Database tool: Firebase

Firebase can match the real-time, lightweight, easy-to-understand nature of the tagging tool to React. Real-time databases are a core feature of Firebase. It can access the database securely from the client code directly. Instead of the usual HTTP requests, Firebase's real-time database uses a data synchronization mechanism. Whenever the data changes, any connected device receives an update at a millisecond rate. It provides a collaborative and immersive experience without having to write network code. (Google, 2020.)

If web development does not require complex data structure relationships but is flexible, lightweight, and easy to understand, Firebase is a good choice. NoSQL is flexible for data storage. In addition to this formalized approach to external tables, users can de-formalize external data directly into the original dataset to improve query efficiency. NoSQL has no concept of strong coupling, and it can delete any data at any time. Low latency read/write speed and fast application response can greatly improve user satisfaction. (Google, 2020.)

3. Research Method

In this section design science research is introduced to answer this study's research question. This section started with a literature review, proposed a design for an annotation tool, and then made the model and constructs according to the relationship between the components. Following the general steps of design science research, the different stages of design science research are presented completely.

3.1 Design science research

Design science research in information systems aims to provide valuable information for researchers to learn, instruct, analyze, and report design science research in information systems and those interested in design science research (Hevner & Chatterjee, 2010). The motivation for design science research is the intention to improve the current state of practice and existing research knowledge by developing an artifact and/or design construction process (Simon, 1996). Design science research is using design, analysis, evaluation, and reflection to establish a lack of information. It is valuable to use existing knowledge to build a design that helps researchers to recognize the degree of missing knowledge and the challenges associated with filling in areas of weakness to discover incomplete knowledge in a new field of design. (Hevner & Chatterjee, 2010.)

Knowledge is generated and acquired through action (Hevner & Chatterjee, 2010). Design science research work focuses on artifact creation, it contains two key activities that improve the action and understanding of information systems, one is developing new understanding by building new or creative artifacts (items or processes) and the other is presenting an overview of the use and/or output of the artifact with analysis and inference. A philosophy of design science research is a way to contribute to knowledge. The difference between design research and design science research is primarily about using design as a research method or technique. (Hevner, 2007.)

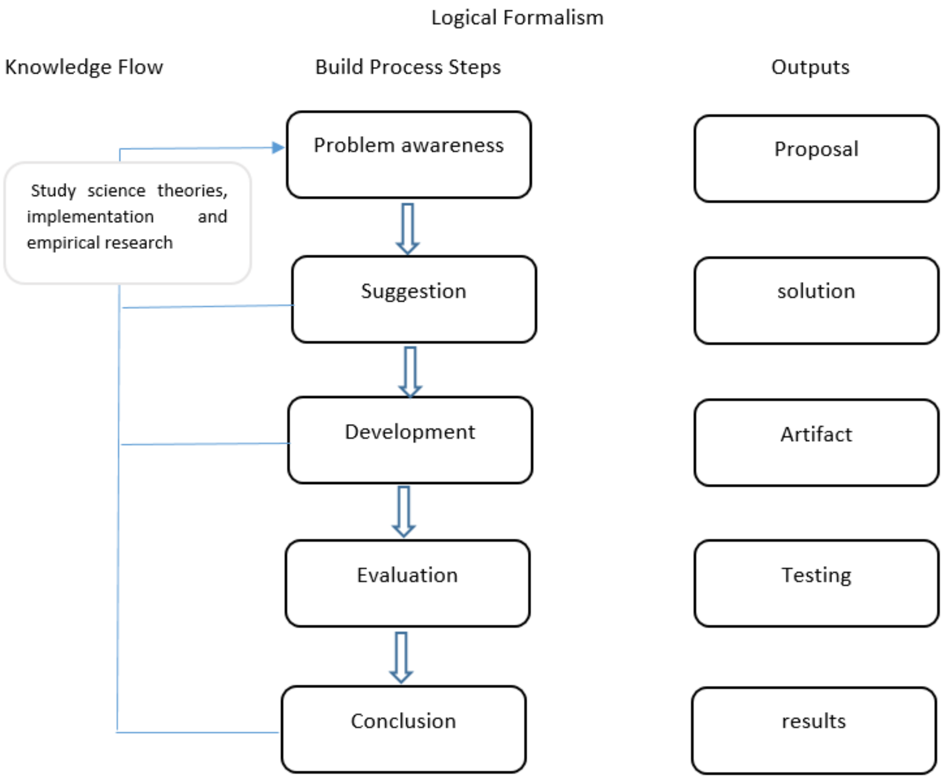


Figure 4. Design Science Research Process Model.

The implementation of a computable design process model is shown in Figure 4. It is a template of the general cycle practiced by design science research institutes. (Takeda, Veerkamp, Tomiyama, & Yoshikawam, 1990.) The next section explains the steps in Figure 4.

Problem awareness: Awareness of an interesting research issue can come from several sources, like new market innovations or the recognition of challenges references. Reading in related disciplines also may offer opportunities for introducing new findings to research areas. The kinds of questions involved with design science research are mostly ways to solve problems instead of questions answered by an explanation. At this step, researchers consider the measures used to determine the research effort's results. The outcomes of this step are formal or informal suggestions for a new research project. Suggestion: After the design of the plan, the suggestion process is based on an understanding of the issue. The suggestion is an innovative step in which, according to the configuration of existing or new and existing components, new functionality is planned. (Hevner & Chatterjee, 2010.) Preliminary design and performance of models based on the concept are part of the plan. Development: At this step, the initial design is further developed and implemented. (Gregor & Jones, 2007.) Many types of artifacts can be created, starting from design theories to ideas, models, methods, or instantiations (March & Smith, 1995). A specialist framework that introduced new ideas regarding human thought in a field of interest may require the creation of software, likely using specialized packages or tools. Implementation is not necessary to require innovation beyond a given artifact's functional state; the innovation is essential in the design. (Gregor & Jones, 2007.) Evaluation: After completion of development, the artifact is evaluated to requirements that are often implicit and sometimes explicit in the proposal (problem awareness phase). The predicted activity and effect of the artifact are presumed to be measured using an evaluation method consistent with the evaluation requirements (Venable, Pries-Heje, & Baskerville, 2016).

Conclusion: This step can be the end of a research cycle or the end of a particular research project. The reported study is presented correctly and its contribution to information is strongly demonstrated. (Hevner & Chatterjee, 2010.)

3.2 Applying design science research

According to the general cycle of design science research, this section reports how the design science research was applied in this study. This new annotation tool was designed by analyzing the biggest obstacle in the labeling process and tested the user's understanding of annotation precision.

3.2.1 Awareness problem and proposal

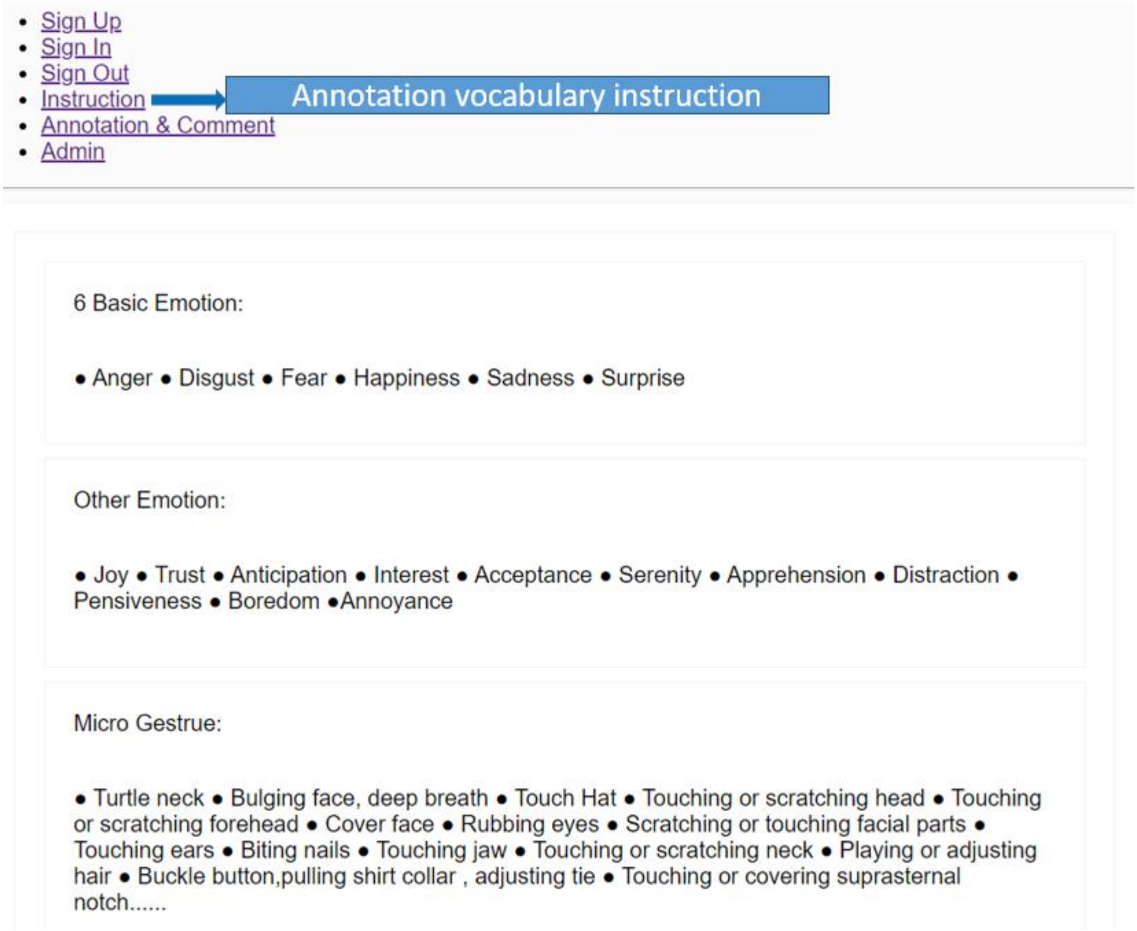
Previous research literature suggests that a typical annotation approach involves having more than one person independently perform the same task until reaching a consensus on an answer. A repeated explanation is the main method to learn the process and explain a relatively difficult behavior. (De Amorim et al., 2017.) A new artifact was developed to collect the new workers' annotation that they interact to performance in this study. This design was inspired by locally experienced computer visual researchers in Oulu. This new tool has an overall list so that crowd workers can quickly compare their annotations to each other.

This annotation tool is a lightweight Web, interactive annotation tool for people to share, comment, and vote of opinions on micro-level human behavior labeling. It is designed to allow users to freely input basic emotion, micro-gesture, comment, and vote, the annotation from a non-professional perspective, showing what workers think, and encouraging workers to contribute the improvement of annotation results, not just the analysis of annotation results. This tool has an overall list so that crowd workers can quickly compare their annotations to others. It is a platform for learning and training. The user interface is designed to enhance interaction and communication. Simple annotation results do not reflect the user's learning process of human behavior. Comments and votes come from different users and gained a diverse understanding of human behavior. It increased users' interest in annotation analysis. All comments served as examples for new user learning and training.

This tool provides some common terms for users to refer to. If these terms do not apply, users could add new ones. This study design supports a natural language, general vocabulary, and grammar, and collect user phrases. For the user's understanding of low and high privacy of human behavior, It requires a high frequency of discussing, especially on complex issues, rather than the boring process that extracts keyframe from a full video and working separately. This tool design made the user more concerned with analyzing a single label result rather than completing the task. It effectively reduced the workload.

3.2.2 Suggestion and solution

This annotation tool UI is designed according to the annotation, comment, and vote feature. The user landing page is shown in Figure 5. It includes the sign-up, sign in, sign out, instruction, annotation, and comment interface. The two main features are comment and annotation. After signing in, users can add a new video to mark any part of a video on the Annotation &Comment page, and then the generated label is automatically displayed on the list for other users to comment on.



6 Basic Emotion:

● Anger ● Disgust ● Fear ● Happiness ● Sadness ● Surprise

Other Emotion:

● Joy ● Trust ● Anticipation ● Interest ● Acceptance ● Serenity ● Apprehension ● Distraction ● Pensiveness ● Boredom ●Annoyance

Micro Gestrue:

● Turtle neck ● Bulging face, deep breath ● Touch Hat ● Touching or scratching head ● Touching or scratching forehead ● Cover face ● Rubbing eyes ● Scratching or touching facial parts ● Touching ears ● Biting nails ● Touching jaw ● Touching or scratching neck ● Playing or adjusting hair ● Buckle button,pulling shirt collar , adjusting tie ● Touching or covering suprasternal notch......

Figure 5. Instruction page.

Commonly used annotation terms are shown on the instruction page in Figure 5. It is not compulsory to use. If none of these terms is appropriate for a particular situation, a new one can be added to the comment.

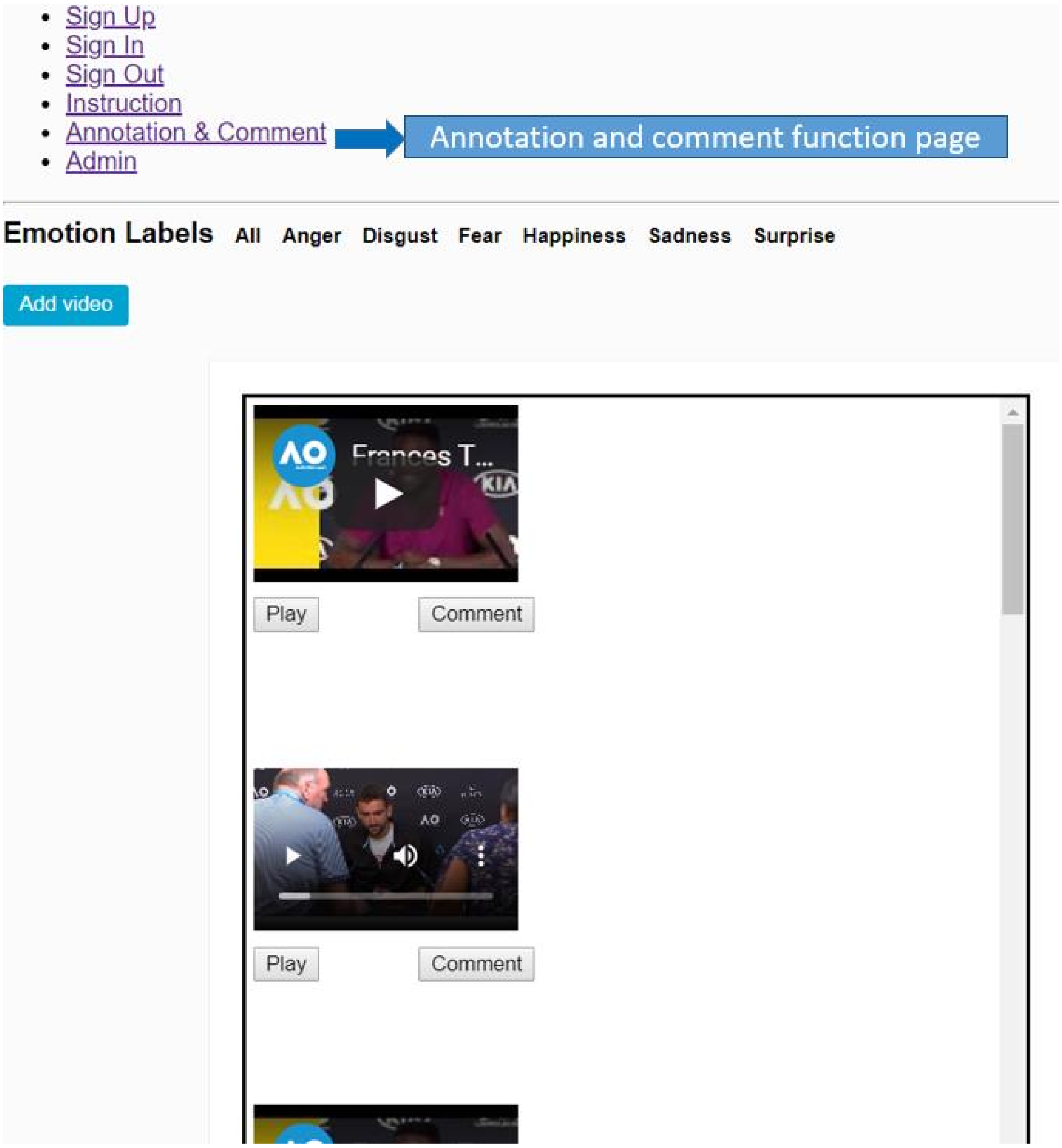


Figure 6. Annotation and comment page.

The administrator can manage user accounts, videos, annotations, and comments on the Admin page. In the Annotation & Comment page, the user can add a new video and annotation to let others review as shown in Figure 6. A list of annotated videos is displayed on this page, it provides video for users to choose according to emotional categories. To add new videos, click on the "Add video" button. Users can play videos and select which ones they are interested in. The comment button is to view existing comments and add new ones.

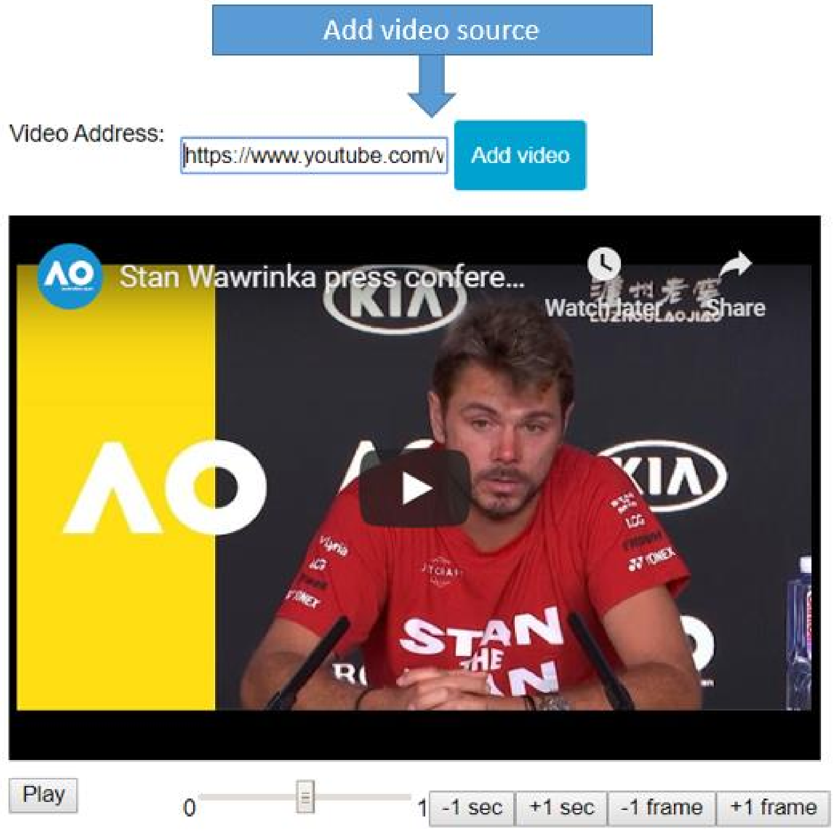
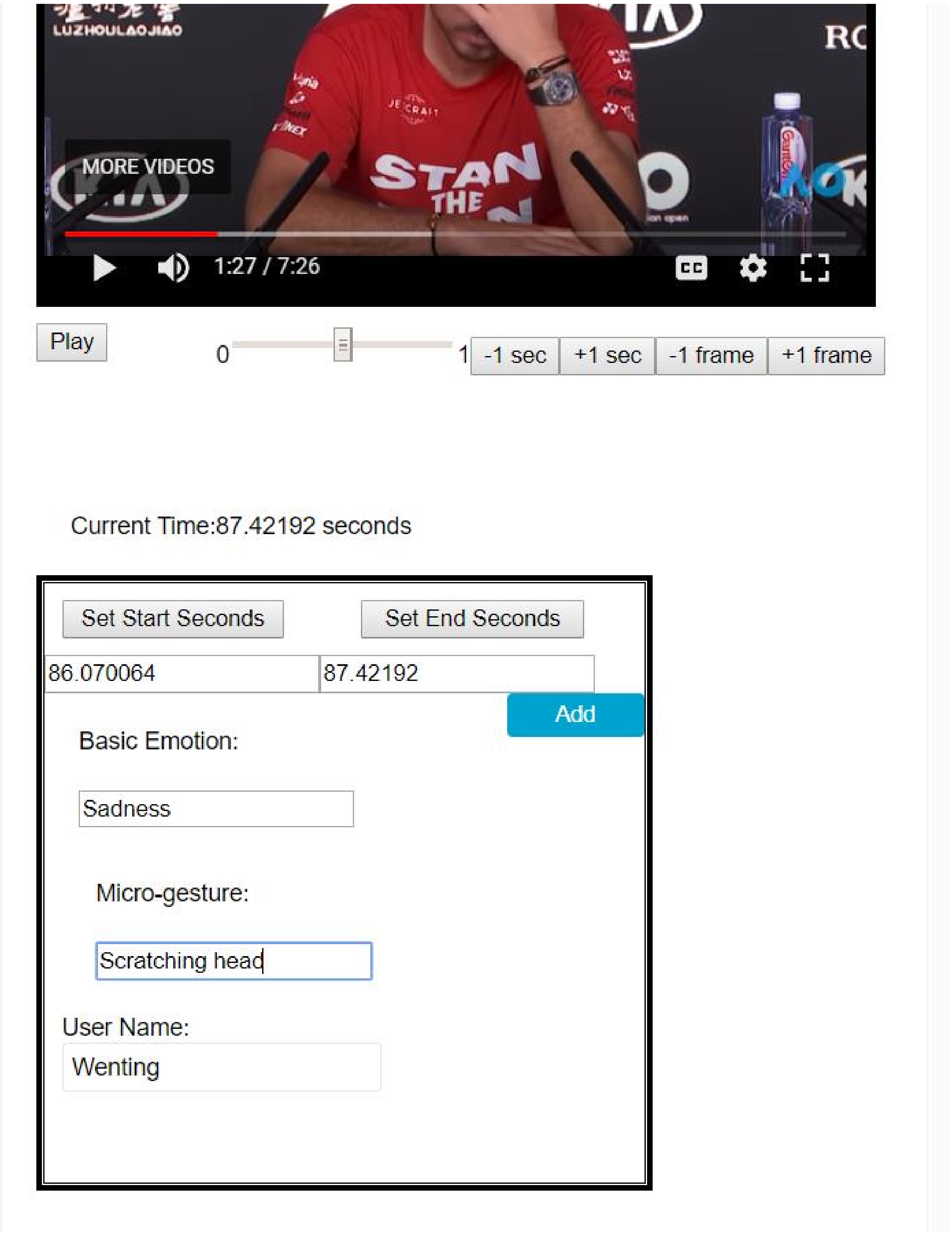


Figure 7. Add a new video.

After click on the “Add video” button, then copy the video address, and click “Add video” as shown in Figure 7, the video will automatically display and show the annotation function.



Current Time:87.42192 seconds

86.070064

87.42192

Basic Emotion:

Sadness

Micro-gesture:

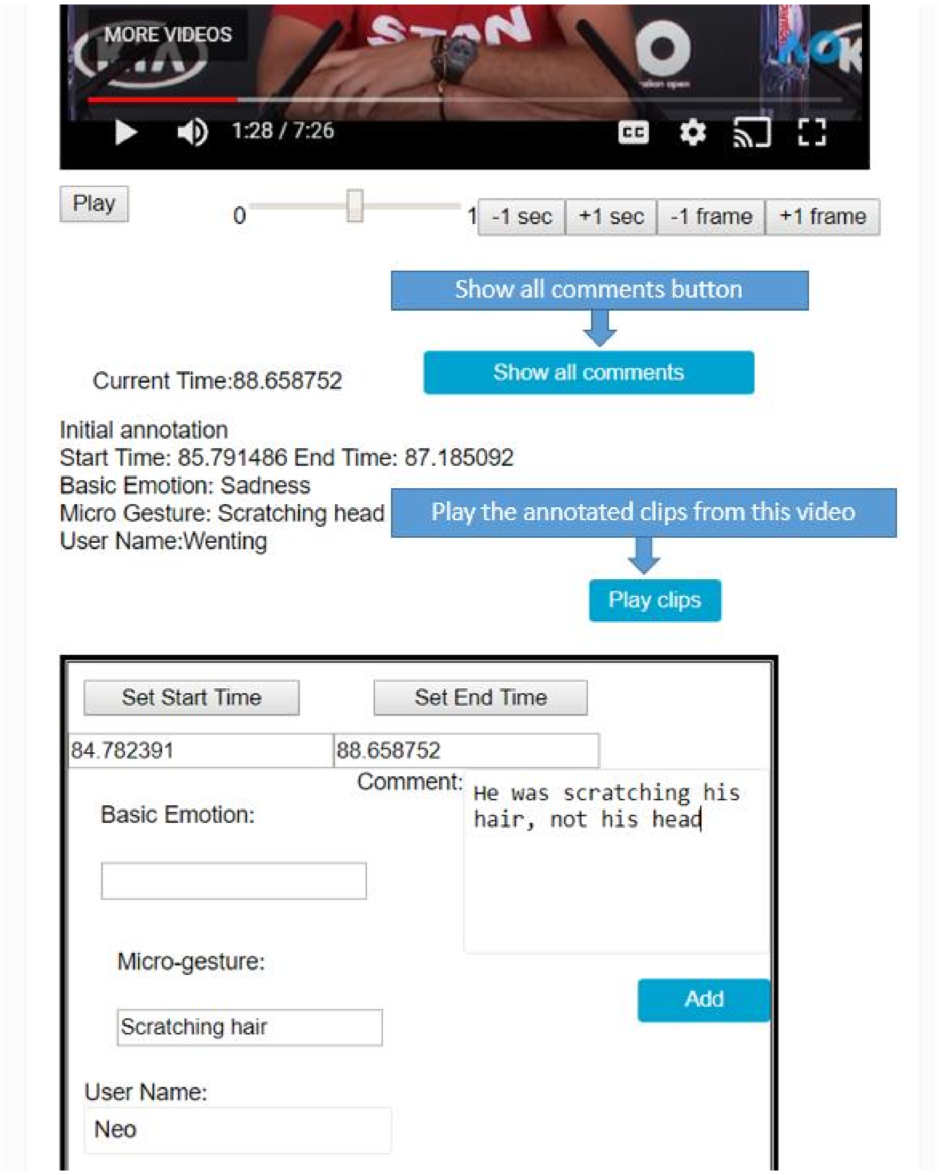
Scratching head

User Name:

Wenting

Figure 8. The annotation process.

A selected video is marked by the user first time as shown in Figure 8 after click "Add" the annotated video is displayed on the video list for other users to review and comment. That is the initial annotation. Annotations time is automatically converted to seconds, and adjusted by 1 second or 1 frame. The annotation functions are to mark the start seconds and end seconds of emotion and micro-gesture.



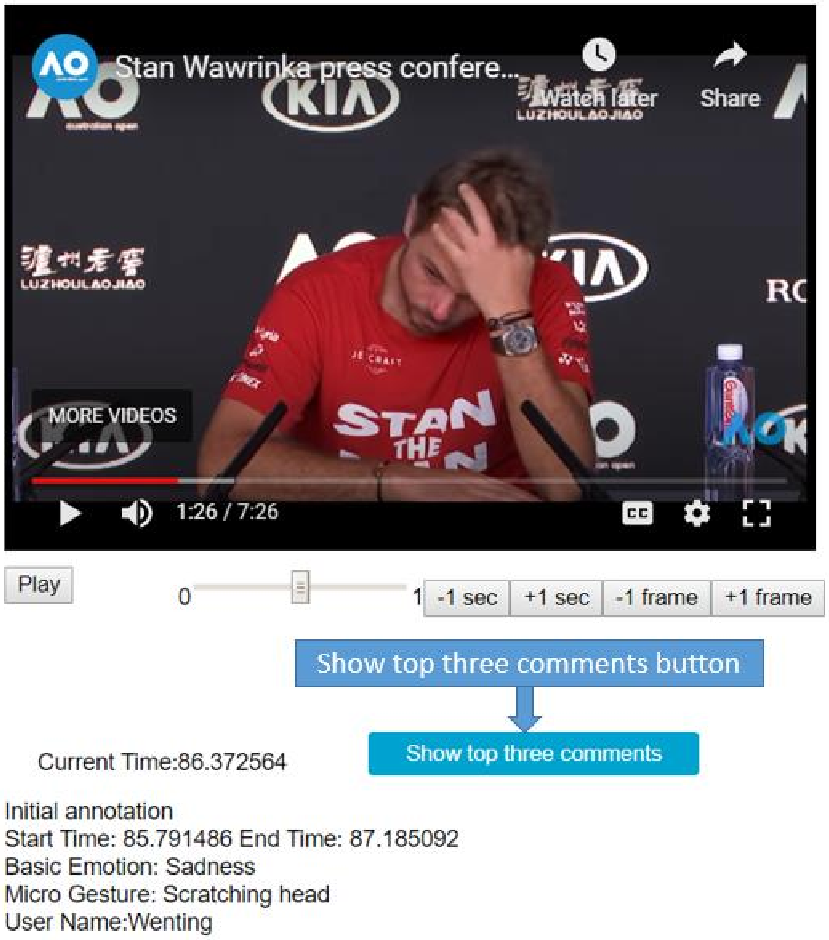
Current Time:88.658752

Initial annotation
Start Time: 85.791486 End Time: 87.185092
Basic Emotion: Sadness
Micro Gesture: Scratching head
User Name:Wenting

84.782391
88.658752
Comment: He was scratching his hair, not his head
Basic Emotion:
Micro-gesture: Scratching hair
User Name: Neo

Figure 9. The annotation process.

When the user selects a video from the video list and clicks the “Comment” button, it can play the initial annotation video clips and read comments, annotate, and fill comments as shown in Figure 9. In this time the annotation and comment will be displayed on the comment list. The comment context part is the explanation of the marked label by natural language. When clicking the “Show all comments” button, users have an overall comment list which enables them to quickly compare opinions. And the button will automatically switch to the “Show top three comments” button.

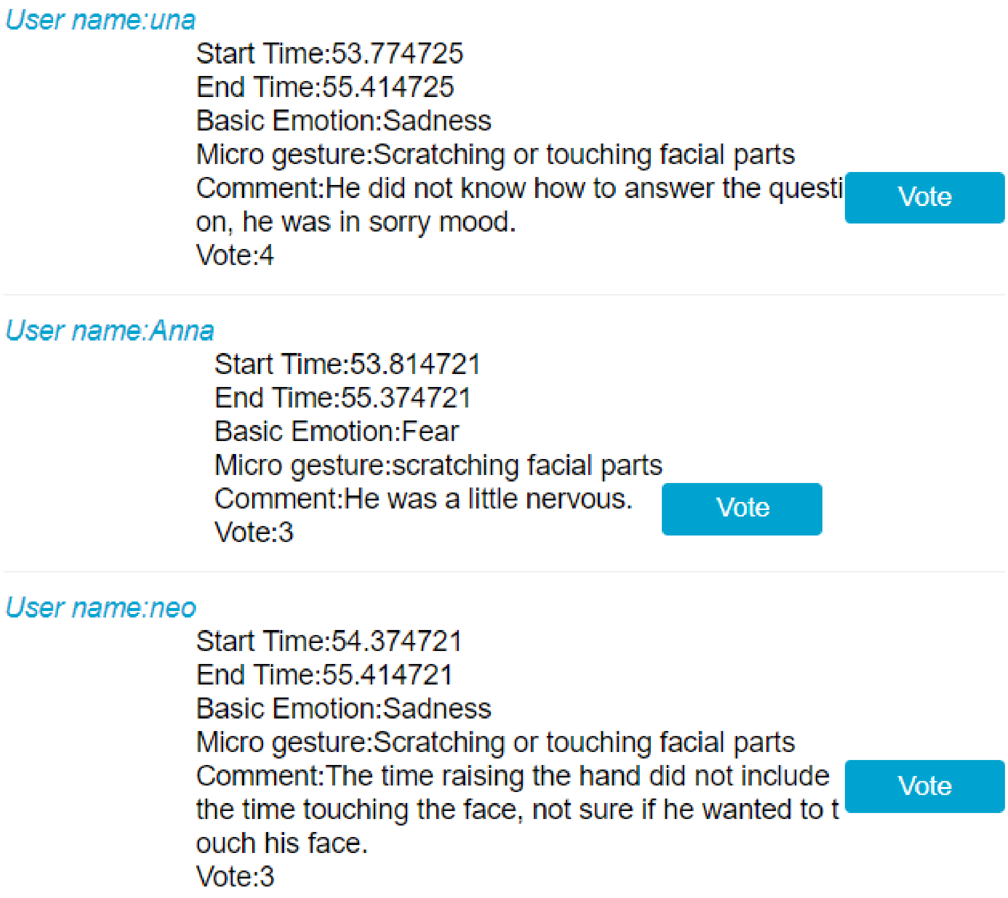


Current Time:86.372564

Initial annotation
Start Time: 85.791486 End Time: 87.185092
Basic Emotion: Sadness
Micro Gesture: Scratching head
User Name:Wenting

Figure 10. The top three comments display button on the comment page.

The top-three comments button prevents comments from being displayed too long, and it makes users more focused on the top three comments as shown in Figure 10.



User name:una

Start Time:53.774725
End Time:55.414725
Basic Emotion:Sadness
Micro gesture:Scratching or touching facial parts
Comment:He did not know how to answer the questi
on, he was in sorry mood.
Vote:4

User name:Anna

Start Time:53.814721
End Time:55.374721
Basic Emotion:Fear
Micro gesture:scratching facial parts
Comment:He was a little nervous.
Vote:3

User name:neo

Start Time:54.374721
End Time:55.414721
Basic Emotion:Sadness
Micro gesture:Scratching or touching facial parts
Comment:The time raising the hand did not include
the time touching the face, not sure if he wanted to t
ouch his face.
Vote:3

Figure 11. Comment list and voting function.

The comments and votes from different users on an event are shown in Figure 11. The comment can be voted if users agree. Labels with a majority of votes are more valuable for beginners to learn.

The operation of this tool is simple, and the feature does not set limits for quality control. Vote function is effective to recognize and filter out noise by comparison.

4. The study, implementation, and empirical research

In this section, the first part reports the development process, including some part code. The second part reports an experimental method to test and evaluate whether the tool can help users accurately annotate.

4.1 Development and artifact

Based on previous research Web annotation tools need to easily port annotation results from videos to popular platforms, and the code needs to be easy to create module tasks (Park et al., 2012). This part reports the implementation code. The development process shows that React code is highly readable, easy to be maintained, and extended.

4.1.1 Web: React

According to the concept of the development of React, the tool development process followed the template by React recommendation (React, 2020).

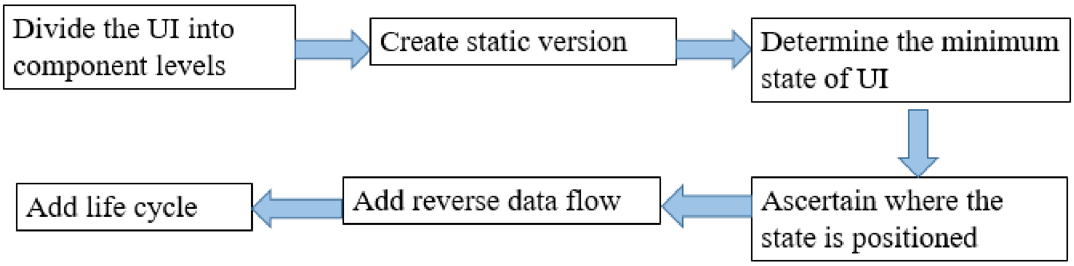

Divide the UI into component levels
Create static version
Determine the minimum state of UI
Ascertain where the state is positioned
Add reverse data flow
Add life cycle

Figure 12. The development process.

React development mode presents in Figure 12 (React, 2020). Its unique idea leads developers on how to build an application. The components are divided into high and low levels after components are determined by function. Then the data flow is identified and classified which data type in the component. Finally, the reverse data flow and life cycle are added.

Divide the UI into component levels: The whole UI of the new tool was divided into six parts. Each component was defined by the single function concept or object. The component was only responsible for one feature. If a component was responsible for more functionality, it was broken down into smaller components, like the comment component was broken into several small components. Their parent component was the APP component, it was responsible for the interaction of the other child components.

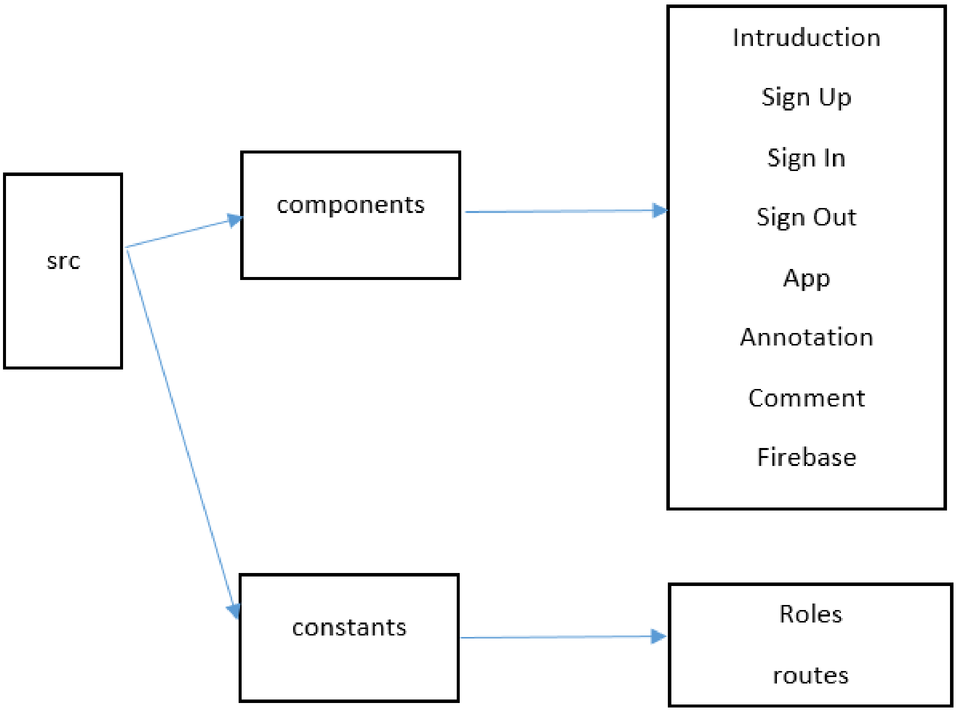


Figure 13. The SCR development architecture.

The framework of this Web app is shown in Figure 13. Under the SRC file is the components file and the constants file. Routes and databases are also considered as components. They are directly used by Import. Then components are to be divided into different levels.

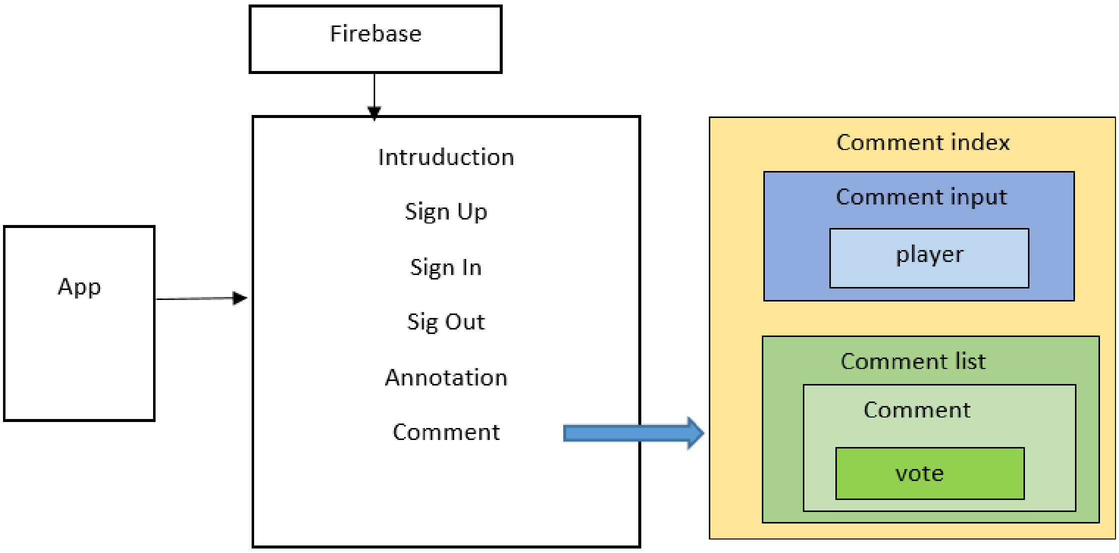


Figure 14. Parent-child relationships in components.

The relationship between the components is shown in Figure 14. The APP component controls all other components. The other functional components are all child components and ultimately imported into the APP component for rendering. The comment components are divided into the player, input of comment, the list of comments, the comment display, the voting function, and the number counter.

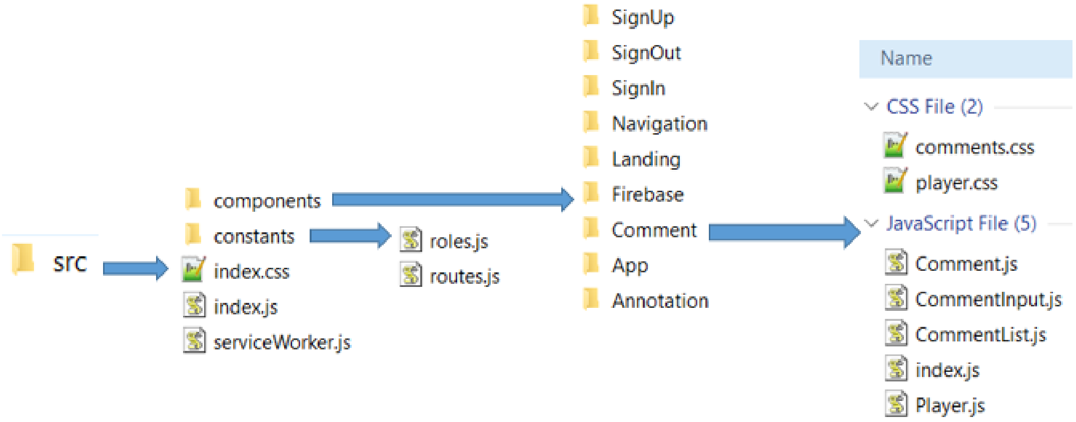


Figure 15. The components file structure.

The structure of the development file is presented in Figure 15. It is easy to reuse each component by import. The data flow passes via props.

Create a static version: This tool started by building a UI with an existing data model without interactive features. It is necessary to keep the rendering UI separate from the added interaction because when writing a static version of an application, one needs to write a lot of code without much interaction detail. But there is a lot of detail to consider when adding a few codes to interactive functionality. (React, 2020.)

The App component was written as a static version first, such as static routing. Static routing was routing configuration defined before application running. This system started to run, loaded the configuration, and built the application routing table. Once the system receives a request, it is applied in the routing table to find out the corresponding page or processing methods according to the address. This application was built from the top which means writing higher-level components first, like ‘App’, and ‘Index’.

App code:

```
import React, { Component } from 'react';
import {
  BrowserRouter as Router,
  Route,
} from 'react-router-dom';
import Link from 'react-router';
import Navigation from '../Navigation';
import LandingPage from '../Landing';
import SignUpPage from '../SignUp';
import SignInPage from '../SignIn';
import Instruction from '../Instruction'
import AnnotationPage from '../Annotation';
import * as ROUTES from '../../constants/routes';
import ReactPlayer from 'react-player';
import CommentsPage from '../Comment';
const App = () => (
  <Router>
        <div>
          <Navigation />
          <hr />
          <Route exact path={ROUTES.LANDING} component={LandingPage} />
          <Route path={ROUTES.SIGN_UP} component={SignUpPage} />
          <Route path={ROUTES.SIGN_IN} component={SignInPage} />
          <Route path={ROUTES.SIGN_OUT} component={SignOutPage} />
          <Route path={ROUTES.ANNOTATION} component={AnnotationPage} />
          <Route path={ROUTES.INSTRUCTION} component={Instruction} />
          <Route path={ROUTES.COMMNENT} component={CommentsPage} />
         </div>
  </Router>
);
export default App;
```

Rout code:

```
export const LANDING = '/';
export const SIGN_UP = '/signup';
export const SIGN_IN = '/signin';
export const SIGN_OUT = '/signout';
export const ANNOTATION = '/annotation';
export const COMMNENT = '/comment';
```

Index code:

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './components/App';
import * as serviceWorker from './serviceWorker';
import Firebase, { FirebaseContext } from './components/Firebase';

ReactDOM.render(
  <FirebaseContext.Provider value={new Firebase()}>
    <App />
  </FirebaseContext.Provider>,
  document.getElementById('root'),
);

serviceWorker.unregister();
```

Since this tool has built a static version, these components only provide the render method for rendering. Once data changes, the system calls ‘reactDom.render’ again, the UI is updated accordingly. It is easy to see how and where the UI is updated. The data rendering section can be updated at any time. For example, if the user selects a video to play, then he paused to get the current time as the starting time for an annotation.

```
class Annotation extends Component {

……

 setStartTimeText= event => {

          this.onPause();

          this.startTimeText = String(this.getCurrentTime());

…….

  }

  render =() => {

    const { url, playing, controls,startTimeText, ……loading } =
this.state

    return (

      <div className="setStartTimeArea">
```

```
         <button                   className="annotationControlButtons"
onClick={this.setStartTimeText} >Set Start Time</button>
</div>
……
}
```

After clicking the start time button, the ‘setStartTimeText’ function performed the logical operation, and the returned result is updated and rendered.

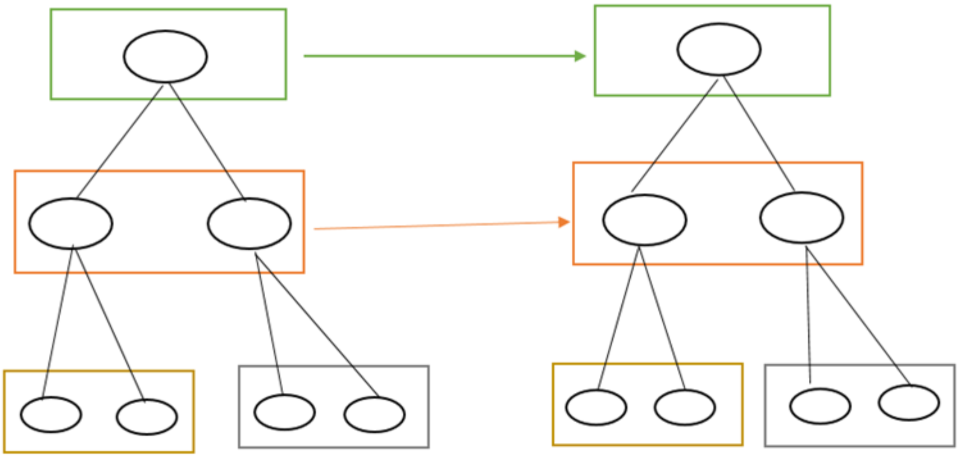


Figure 16. Parent-child relationship of the render hierarchy.

Figure 16 presents the different components and hierarchical rendering mechanisms. Each color belongs to the same level. Once the data of one of the lower components is changed, the changed parts are compared layer by layer, and only the changed parts are re-rendered. This is the virtual DOM, which ensures that only real DOM manipulation is done on the parts of the interface that change. (React, 2020.)

Reducing render is one of the keys to improving project performance. For functional components, the call to the render UI is triggered when the state value changes. If there is no change in the state, the system would call ‘setState’ to trigger a render. Because the React is a class of components inherited, once the parent container re-render, the component's render is called again. One way of React optimization is to less render. Therefore, to avoid unnecessary overhead, it places specific state values at a lower level or component. Since each state update would trigger a new render call, fewer state updates result in fewer calls to render. This way merges state updates to avoid having to do each state update after a state change. (React, 2020.)

Determine the minimum state of UI: To make the UI interactive, it needs the ability to trigger changes to the underlying data model. React does this by implementing ‘state’. Only the minimal set of variable ‘state’ required by the application is retained, and all other data is computed from them. First, determine whether the data is ‘state’ or ‘props’. The basis of judgment is: If this data is passed by the parent component via ‘props’, it is not ‘state’. If the data remains constant over time, it is not ‘state’. If the data is calculated based on other ‘state’ or ‘props’, it is not ‘states’. (React, 2020.)

Table 1. State and props data.

state	**props**
Basic emotion	Current time
Micro gesture	Start time
Comment context	End time
User name	Vote number
Video URL	Annotation list
	Comment list

The ‘state’ and ‘props’ are classified as shown in Table 1. The ‘state’ and ‘props’ are usually confused. There are many materials and examples to discuss the differences and commonalities between them. Overall, using ‘props’ and ‘state’ are keys to the React, they depend on the way of interpretation, and the effects are directly rendered on the UI. Most components get the data from the ‘props’ property and render it. Sometimes the component has to respond to input, interact with the server, and in those cases, it has to use ‘state’. The official document advice on React is: keep components as stateless as possible. The state separates from the business logic, reduces redundancy, and keeps the component's single responsibility as much as possible. The ‘state’ represents the internal state of the component itself, it is private and completely controlled by the component. React recommendation is to build ‘stateless’ components to render data, build the ‘stateful’ component to interact with users and services on top of that, and pass the data to the stateless component via ‘props’. The ‘state’ contained the most original data. ‘Props’ are the way from which the parent component passes data to the child component. (React, 2020.)

```
import React, { Component } from 'react'
import Comment from './Comment'

class CommentList extends Component {
   static defaultProps = {
   comments: []
        }
  render() {
    return (
      <div>
        {this.props.comments.map((comment,      i)      =>      <Comment
comment={comment} key={i} />)}
      </div>
          )
  }
}

export default CommentList
```

This Commentlist component code shows how to use ‘props’ to pass comment data. The Commentlist component converts the array of comments into a list, traverses the element with the ‘map’, uses the ‘key’ to update the location, and reduces the performance overhead.

Ascertain where the state is positioned: As the data flow in React is one-way and passes down the component hierarchy, first found all the components that the ‘state’ rendered, their co-owner was comment ‘index’, it owned the ‘state’.

Comment index code:

```
import React, { Component } from 'react'
import './comments.css'
import CommentInput from './CommentInput'
import CommentList from './CommentList'

class CommentsPage extends Component {
   constructor (props) {
    super(props)
    this.state = {
      comments: [],
    }
  }

  handleSubmitComment (comment) {
     console.log(comment)
     this.state.comments.push(comment)
     this.setState({
      comments: this.state.comments
    })
  }

  render() {
    return (

          <div className='wrapper'>
            <CommentInput
onSubmit={this.handleSubmitComment.bind(this)} />
            <CommentList comments={this.state.comments}/>

          </div>
    )
  }
}

export default CommentsPage
```

Part of the CommentInput code:

```
class CommentInput extends Component {

  constructor(props){
        super(props);
        this.state = {
            newopinion: '',
            newgesture:'',
                .......
        };
  }

    handleSubmit = event =>  {
      if (this.props.onSubmit) {
      this.props.onSubmit({
```

```
                    startTimeText: this.startTimeText,
                    endTimeText: this.endTimeText,
                    newopinion:this.state.newopinion,
                    newgesture:this.state.newgesture,
                    username: this.state.username,
                    content: this.state.content,
                    votes: '0',
                    })
    }
        this.props.firebase.commentTexts().push({
                    startTimeText: this.startTimeText,
                    endTimeText: this.endTimeText,
                    newopinion:this.state.newopinion,
                    newgesture:this.state.newgesture,
                    username: this.state.username,
                    content: this.state.content,
                    votes: '0',
    });
    this.setState({ newopinion: '', newgesture:'', content: ''......})
  }

    render =() => {
    const {newopinion,newgesture, content,username......} = this.state

    return (
       ……
        <div className='comment-field-button'>
             <button
                   onClick={this.handleSubmit.bind(this)}>
                   Add
             </button>
       ……
        )
      }

    }
```

This ‘CommentInput’ component code shows the original input data from the user. These data belong to the ‘state’. The start and end times are passed through the player, vote number is computed by clicking function, these data are ‘props’.

Add reverse data flow: Every time the user changes the value of the comment, it needs to change ‘state’ to reflect the user's current input. Since ‘state’ only is changed by the component that owns them, the ‘handleOpinionChange’ event is used to monitor changes in the user input. Then the callback function calls ‘setState’ to update the application.

Part of the CommentInput code:

```
class CommentInput extends Component {
      ……

      handleOpinionChange (event) {
        this.setState({
          newopinion: event.target.value
        })
      }
        render =() => {
        const {newopinion,......} = this.state

    return (
       ……
```

```
        <span className='setNewOpinionArea'>Basic Emotion:</span>
              <div className='setNewOpinionArea'>
                    <input value={this.state.newopinion}

onChange={this.handleOpinionChange.bind(this)}/>
               </div>
        ……
         )
       }
     }
```

The ‘handleOpinionChange’ function code shows when ‘setState’ is called, React merges the ‘new opinion’ object into the current state and then called ‘setState’ separately to update the object separately.

Add life cycle: Because this tool needs to display the current annotation, it initializes ‘this.state’ with an object that contains the current annotation and then updates the ‘state’. React calls the component's render method and updates the DOM to match the output of the annotation rendering. After the output of the annotation is inserted into the DOM, React invokes the ‘ComponentDidMount’ lifecycle method. The ‘AnnotationInput’ component updates the UI by calling ‘setState’. Once React calls ‘setState’, it realizes that the ‘state’ has changed, and then calls the render method again to determine what is displayed on the page. This time ‘this.state.labelTexts’ is different, rendering the updated annotation. Once the annotation component is removed from the DOM, React invokes the ‘componentWillUnmount’ lifecycle method, and it stops.

Part AnnotationInput code:

```
const AnnotationInput =() =>(
   <div>
      <Annotator/>
    </div>
)

class AnnotatorBase extends Component {
 …….
  componentDidMount(){
      this.setState( {loading: true});
      this.props.firebase.labelTexts().on('value', snapshot =>{
         const labelTextObject = snapshot.val();

         const labelTextsList = Object.keys(labelTextObject).map((key)
=> {return {
             uid: key,
             startTimeText: labelTextObject[key].startTimeText,
             endTimeText: labelTextObject[key].endTimeText,
             newopinion: labelTextObject[key].newopinion,
             microGesture: labelTextObject[key].microGesture,
             username: labelTextObject[key].username,

             original: labelTextObject[key],
             key: key,
                  }});


           this.setState({
             labelTexts: labelTextsList,
             loading:false,
                  });
          });
  }

  componentWillUnmount(){

          this.props.firebase.labelTexts().off();
  }
 ……
}

const Annotator = withFirebase(AnnotatorBase);
export default AnnotationInput;
```

This modularity and clarity code is easily understandable if it comes to building larger component libraries. As the component is reused, it would significantly reduce the amount of code.

4.1.2 Database: Firebase

Firebase class is the connection between React application and the Firebase API. It is instantiated once and passed to our React application via the React's Context API. (Google, 2020.) The annotation tool Firebase API is defined to connect the Firebase class.

Part Firebase code:

```
class Firebase {
……
  labelText = uid => this.db.ref(`labelTexts/${uid}`);
  labelTexts = () => this.db.ref('labelTexts');
……
}
```

Context code:

```
import React from 'react';

const FirebaseContext = React.createContext(null);

export const withFirebase = Component => props => (
  <FirebaseContext.Consumer>
    {firebase => <Component {...props} firebase={firebase} />}
  </FirebaseContext.Consumer>
);

export default FirebaseContext;
```

Instead of using the Firebase Context directly in the component, each component uses the higher-order component to wrap. The annotator component accesses the Firebase instance via the higher-order component.

```
import { withFirebase } from '../Firebase';

const AnnotationInput =() =>(
   <div>
      <Annotator/>
    </div>
)
class AnnotatorBase extends Component {
```

```
…….
}
const Annotator = withFirebase(AnnotatorBase);
export default AnnotationInput;
```

This Annotator code shows how to wrap the whole component to access Firebase. Rather than using a render prop component, which passes the Firebase instance to the 'AnnotationInput', this way does not need to know about the Firebase instance.

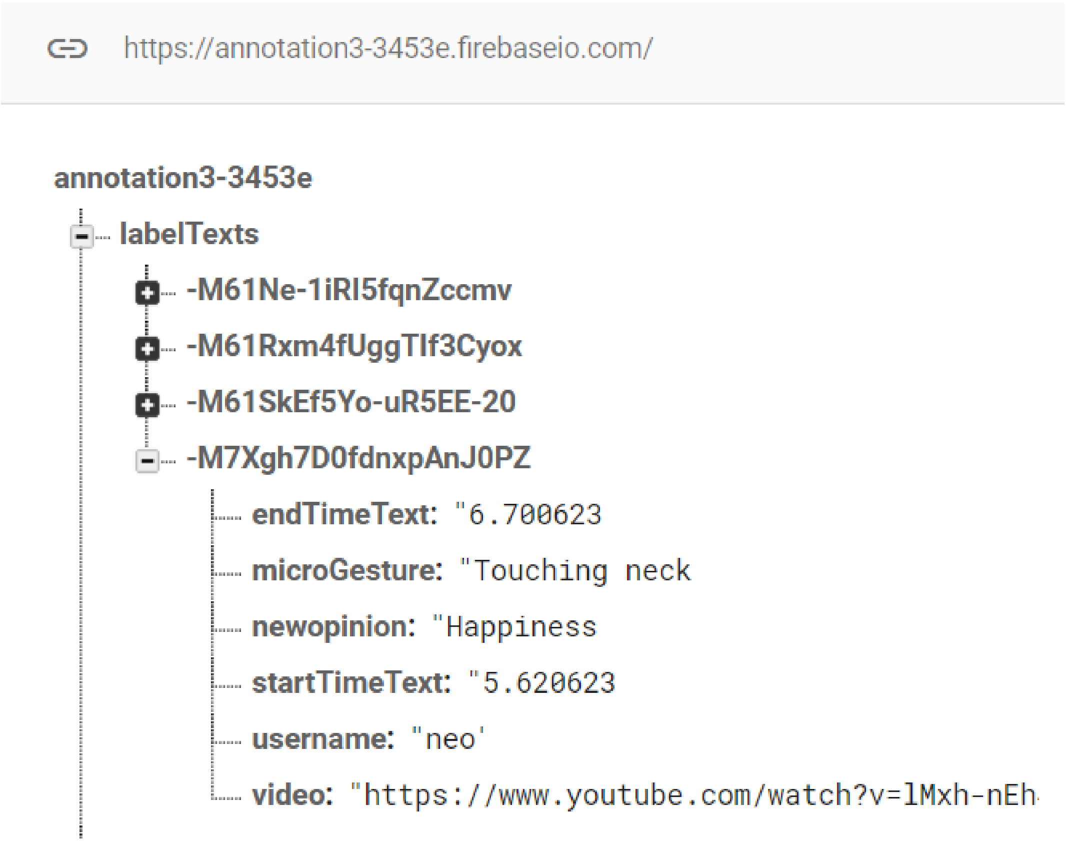
```
handleSubmit = event => {
……

  this.props.firebase.labelTexts().push({
                startTimeText:this.startTimeText,
                endTimeText:this.endTimeText,
                newopinion:this.state.newopinion,
                microGesture:this.state.microGesture,
                username:this.state.username,
                video: this.state.url,
 });

   this.setState({
                startTimeText: '',
                TimeText: '',
                newopinion:'',
                microGesture:'',
                username:'',
  });
event.preventDefault();

  }
```

The tables can be added or reduced to any type directly by the requirement. This process is simple without redoing many operations on the database.



```
https://annotation3-3453e.firebaseio.com/

annotation3-3453e
  labelTexts
    -M61Ne-1iRI5fqnZccmv
    -M61Rxm4fUggTlf3Cyox
    -M61SkEf5Yo-uR5EE-20
    -M7Xgh7D0fdnxpAnJ0PZ
      endTimeText: "6.700623
      microGesture: "Touching neck
      newopinion: "Happiness
      startTimeText: "5.620623
      username: "neo'
      video: "https://www.youtube.com/watch?v=1Mxh-nEh
```

Figure 17. Firebase data on the web.

The database table of labels is shown in Figure 17. Labels' properties are inserted flexibly. There are no relational structures and no primary key. In addition to Firebase's formalized approach to external tables in NoSQL, it can easily add external data directly into the original dataset to improve query efficiency. Low latency read/write speed and fast application response greatly improve development. (Google, 2020.)

4.2 Evaluation and testing

Previous research has shown that voted on the case of the independent annotation is considered agreement. Voting can be used to evaluate the correctness of the responses from the crowd (Yuen et al., 2011). In this study, a new experiment was to recruit people who are interested in participating in and completing the annotation in good faith. The volunteers came from a variety of industries and were between 30 and 40 years old. All initial annotations will be discussed, The results of the crowd generated votes were compared with an expert's data. The expert is a local computer vision expert from Oulu. The video sequence was selected for the reporter interview after the tennis tournament, and the players had obvious emotional characteristics and micro-level gestures when answering.

In the evaluation, ten fresh users took part in annotation and commented on 10 video clips. Finally, 98 valid labels and comments were yielded. The results showed that after voting the frame number of the top three is close to each other, and the mixture of micro gestures and micro emotion can be easily found, the voting can help the fresh worker to find a simple annotation reference.

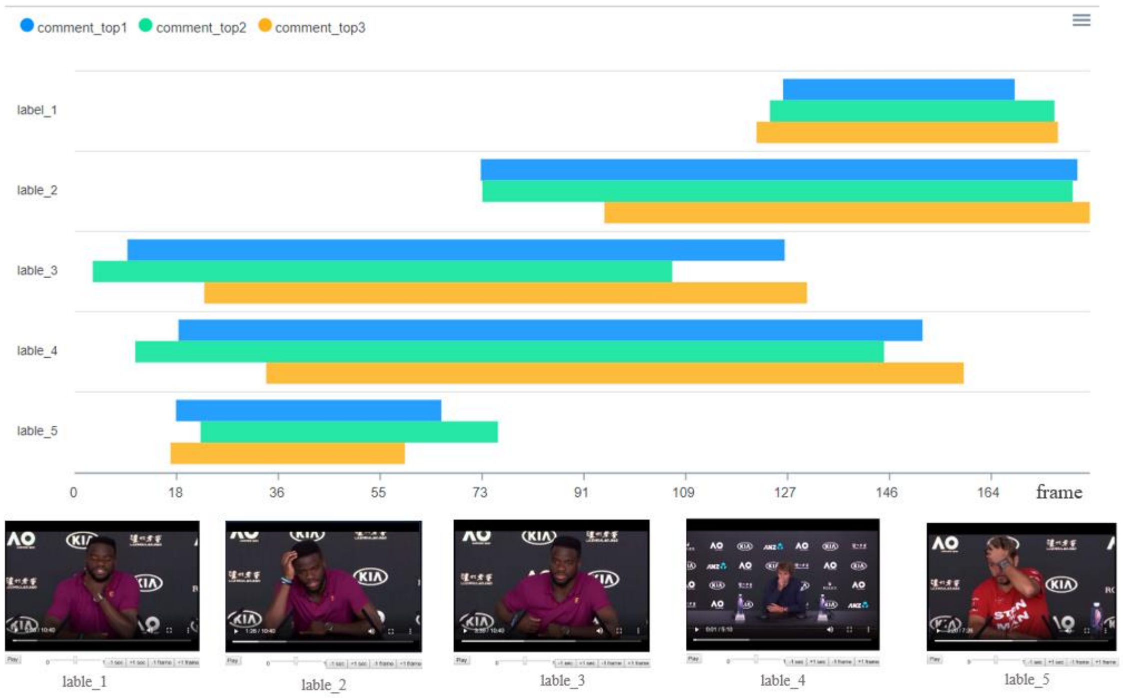
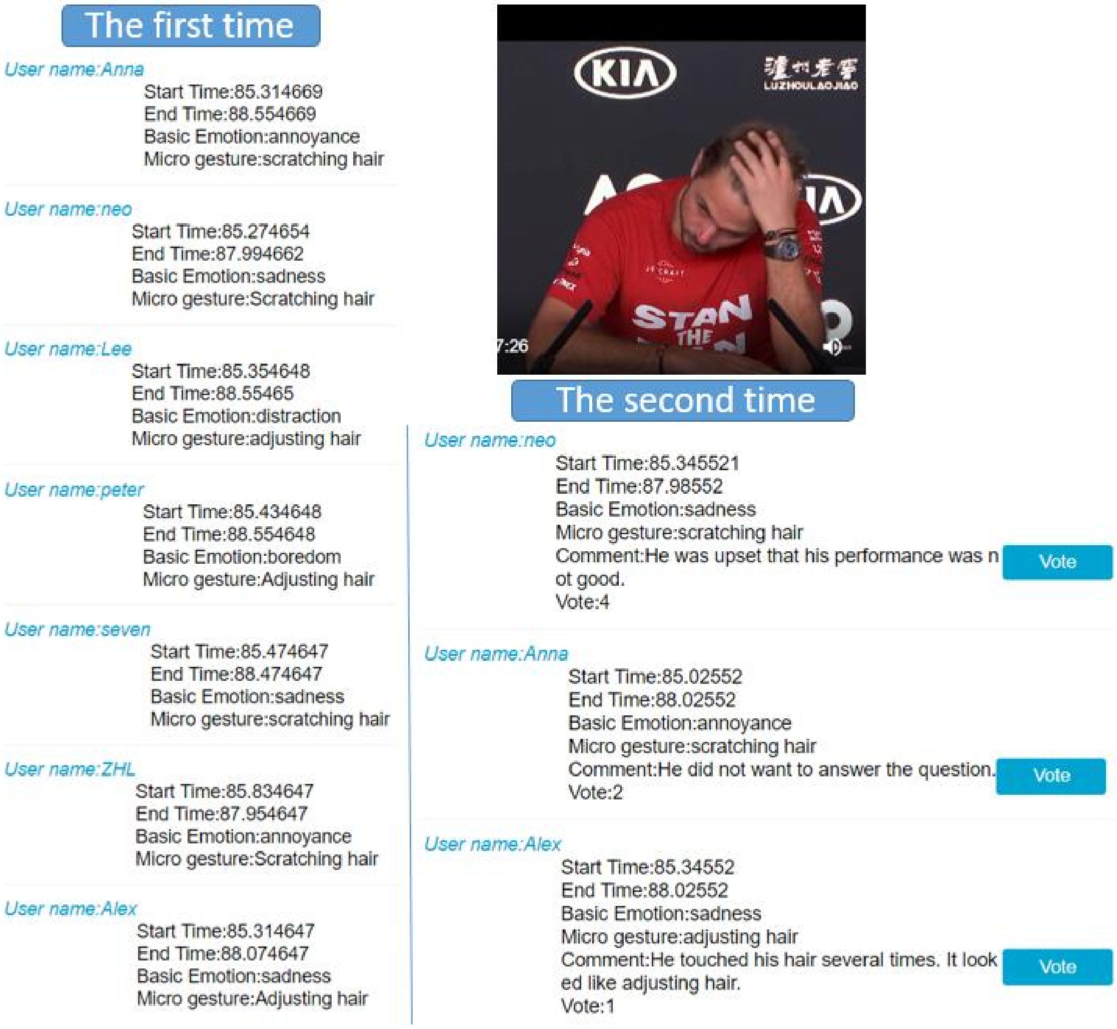


Figure 18. Time label analysis.

Five from ten video clips of the time annotation are shown in Figure 18. Here, the video time is converted into the frames. And a ground truth data labeled by an expert was used to compare with the data by other users. The Precision (Perruchet, & Peereman, 2004) is utilized to evaluate the accuracy of time labeling quantity, which can be computed by Precision = TP / (TP + FP), where TP is true positive (Perruchet et al., 2004), namely the overlap time (annotation) between the user and the expert. FP is false positive (Perruchet et al., 2004), namely the part where the time of the user annotation is inconsistent with the time of the expert annotation. The video frame rate is 25 FPS (frames per second). The range of time precision is between 0.76875 and 0.86439. The average precision is 0.85.



User name:Anna
Start Time:85.314669
End Time:88.554669
Basic Emotion:annoyance
Micro gesture:scratching hair

User name:neo
Start Time:85.274654
End Time:87.994662
Basic Emotion:sadness
Micro gesture:Scratching hair

User name:Lee
Start Time:85.354648
End Time:88.55465
Basic Emotion:distraction
Micro gesture:adjusting hair

User name:peter
Start Time:85.434648
End Time:88.554648
Basic Emotion:boredom
Micro gesture:Adjusting hair

User name:seven
Start Time:85.474647
End Time:88.474647
Basic Emotion:sadness
Micro gesture:scratching hair

User name:ZHL
Start Time:85.834647
End Time:87.954647
Basic Emotion:annoyance
Micro gesture:Scratching hair

User name:Alex
Start Time:85.314647
End Time:88.074647
Basic Emotion:sadness
Micro gesture:Adjusting hair

User name:neo
Start Time:85.345521
End Time:87.98552
Basic Emotion:sadness
Micro gesture:scratching hair
Comment:He was upset that his performance was n
ot good.
Vote:4

User name:Anna
Start Time:85.02552
End Time:88.02552
Basic Emotion:annoyance
Micro gesture:scratching hair
Comment:He did not want to answer the question.
Vote:2

User name:Alex
Start Time:85.34552
End Time:88.02552
Basic Emotion:sadness
Micro gesture:adjusting hair
Comment:He touched his hair several times. It look
ed like adjusting hair.
Vote:1

Figure 19. Comparison with votes and without votes.

As micro emotion and micro gesture, two times annotation on one event were compared in Figure 19. Performance has improved with the vote results. The results after the vote were clear and very close to those of the experts. For the first time, users marked separately, they could not see the results of each other. As such the results were very different, and it took more time to mark then the second time. The second time, users saw the label results of each other. If users agreed with the marked results, they voted for it directly. If they disagreed with the marked results, they marked and commented again. This time users took less time to mark. The user's feedback was that the label was easily marked when other results were shown. Users paid attention to which one was more precise. This method prevented the user from cheating to complete the task. The user's suspicious attitude reduced carelessness, and the review was simple and intuitive.

5. Discussion and conclusion

Through design science research, this study answered the research questions. This study demonstrated the effectiveness of annotation tools for comment and vote. The results showed that users could learn more about data categories based on previous labels results. Labeling by precedent was easier and quicker. It used non-expert crowd intelligence to generate less laborious labels. The votes by various non-experts still produced accurate results. It showed that only a few non-experts working separately are not good at labeling human micro-level behavior. But most votes filter out the noisy judgments to distinguish the fuzziness of emotions, the uncertainty of words, and the inaccuracy of time. Form and feature of quality control would lead to lower communication. However, there are limitations to the experiment in this study. The number of videos is small, and there is not a large enough number of labels to collect. The video selected is limited to an athlete's interview after the tennis tournament.

This study emphasized that the interpretation and description of human behavior requires a lot of implicit knowledge. It had to take a lot of time to recognize the implicit knowledge of human behavior. This annotation tool in this study encouraged workers from different places, languages, and cultures to share their experiences, and explore more possibilities. Comments functionality was also peer-reviewed. It also could be regarded as an analysis update of labels. Each task could be discussed in the order by user requirement, instead just following a set client's pattern. The tool aimed to guide the user to label correctly, not only to complete the label work. Contributing to this tool was essential, to benefit every user. The lightweight tools developed by React and Firebase can be applied to all types of video annotations.

This research analyzed that shared labels to comment and vote was more effective and reliable. Implicit knowledge can be made clear by discussion. This annotation tool provided a way for new users to train. The most important feature of annotation tools for micro-level human behavior was to obtain precise results. Understanding human behavior was a complex and very difficult issue. Our research was the beginning of a series. There are a lot of Web resources available to improve precision and shorten annotation time. We have a few areas of improvement work to develop in the future.

References

Barnard, K., Duygulu, P., Forsyth, D., Freitas, N. D., Blei, D. M., & Jordan, M. I. (2003). Matching words and pictures. *Journal of Machine Learning Research*, *3*(Feb), (pp. 1107-1135).

Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., & Gorrell, G. (2013). Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, *47*(4), (pp. 1007–1029).

Cristani, M., Raghavendra, R., Del Bue, A., & Murino, V. (2013). Human behavior analysis in video surveillance: A social signal processing perspective. *Neurocomputing*, *100*, (pp. 86–97).

Dasiopoulou, S., Giannakidou, E., Litos, G., Malasioti, P., & Kompatsiaris, Y. (2011). A survey of semantic image and video annotation tools. In *Knowledge-driven Multimedia Information Extraction and Ontology Evolution* (pp. 196–239). Springer.

De Amorim, M. N., Segundo, R. M., Santos, C. A., & Tavares, O. D. L. (2017, October). Video Annotation by Cascading Microtasks: a Crowdsourcing Approach. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web* (pp. 49-56).

Enser, P. (2000). Visual image retrieval: seeking the alliance of concept-based and content-based paradigms. *Journal of Information Science*, *26*(4), (pp. 199-210).

Fridlund, A. J. (1997). The new ethology of human facial expressions. *The Psychology of Facial Expression*, *103*.

Gackenheimer, C. (2015). *Introduction to React*. Apress.

Gao, Y., Wang, W.-B., Yong, J.-H., & Gu, H.-J. (2009). Dynamic video summarization using two-level redundancy detection. *Multimedia Tools and Applications*, *42*(2), (pp. 233–250).

GitHub, (2020). *React*. From https://github.com/facebook/react

Google, (2020). *Firebase*. From https://firebase.google.com/

Heggland, J. (2002, September). Ontolog: Temporal annotation using ad hoc ontologies and application profiles. In *International Conference on Theory and Practice of Digital Libraries* (pp. 118-128). Springer, Berlin, Heidelberg.

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In *Design Research in Information Systems* (pp. 9–22). Springer.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, *19*(2), 4.

Izard, C. E. (1997). Emotions and facial expressions: A perspective from Differential Emotions Theory. *The Psychology of Facial Expression*. Cambridge University Press, Cambridge, UK. (pp. 57-77).

Juslin, P., Scherer, K., Harrigan, J., & Rosenthal, R. (2005). *The New Handbook of Methods in Nonverbal Behavior Research.* Oxford: Oxford University Press.

Lai, K., Bo, L., Ren, X., & Fox, D. (2011, May). A large-scale hierarchical multi-view RGB-D object dataset. In *2011 IEEE International Conference on Robotics and Automation* (pp. 1817-1824).

Lewis, M., Haviland-Jones, J. M., & Barrett, L. F. (Eds.). (2010). *Handbook of emotions*. Guilford Press.

Li, Y., Lu, J., Zhang, Y., Li, R., & Zhou, B. (2009). A novel video annotation framework based on video object. In *2009 International Joint Conference on Artificial Intelligence* (pp. 572–575).

Lin, C. Y., Tseng, B. L., & Smith, J. R. (2003, July). VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learning. In *IEEE International Conference on Multimedia and Expo* (pp. 1-2).

MKLab, (2009). *VIA*. From https://mklab.iti.gr/via/

Nowak, S., & Rüger, S. (2010). How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval* (pp. 557-566).

Pantic, M., Pentland, A., Nijholt, A., & Huang, T. S. (2007). Human computing and machine understanding of human behavior: A survey. In *Artificial Intelligence for Human Computing* (pp. 47–71). Springer.

Park, S. (2016). *Computational Modeling of Human Behavior in Negotiation and Persuasion: The Challenges of Micro-Level Behavior Annotations and Multimodal Modeling* (Doctoral dissertation, University of Southern California).

Park, S., Mohammadi, G., Artstein, R., & Morency, L.-P. (2012). Crowdsourcing micro-level multimedia annotations: The challenges of evaluation and interface. In *Proceedings of the ACM Multimedia 2012 Workshop on Crowdsourcing for Multimedia* (pp. 29–34).

Park, S., Shoemark, P., & Morency, L.-P. (2014). Toward crowdsourcing micro-level behavior annotations: the challenges of interface, training, and generalization. In *Proceedings of the 19th International Conference on Intelligent User Interfaces* (pp. 37–46).

Park, S., & Yang, C. M. (2019). Interactive video annotation tool for generating ground truth information. In *2019 Eleventh International Conference on Ubiquitous and Future Networks* (pp. 552–554).

Perruchet, P., & Peereman, R. (2004). The exploitation of distributional information in syllable processing. *Journal of Neurolinguistics*, 17(2-3), 97-119.

Rashtchian, C., Young, P., Hodosh, M., & Hockenmaier, J. (2010, June). Collecting image annotations using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with*

Amazon's Mechanical Turk (pp. 139-147). Association for Computational Linguistics.

React. (2020), *Tutorial: Intro to React*. From https://reactjs.org/tutorial/tutorial.html

Smeulders, A. W., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), (pp. 1349–1380).

Snow, R., O'connor, B., Jurafsky, D., & Ng, A. Y. (2008, October). Cheap and fast–but is it good? Evaluating non-expert annotations for natural language tasks. *In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 254-263).

Spiro, I., Taylor, G., Williams, G., & Bregler, C. (2010, June). Hands by hand: Crowd-sourced motion tracking for gesture annotation. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops* (pp. 17-24). IEEE.

Takeda, H., Veerkamp, P., Tomiyama, T., and Yoshikawam, H. (1990). "*Modeling Design Processes*." AI Magazine Winter: 37–48.

Tani, L. F. K., Ghomari, A., & Tani, M. Y. K. (2019). A semi-automatic soccer video annotation system based on Ontology paradigm. In *2019 10th International Conference on Information and Communication Systems* (pp. 88–93)

Truong, B. T., & Venkatesh, S. (2007). Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1), (pp. 3–8).

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A framework for evaluation in design science research. *European Journal of Information Systems*, *25*(1), 77-89.

Vezzani, R., & Cucchiara, R. (2008). Annotation collection and online performance evaluation for video surveillance: the visor project. In *2008 IEEE Fifth International Conference on Advanced Video and Signal-based Surveillance* (pp. 227–234).

Vondrick, C., Patterson, D., & Ramanan, D. (2013). Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, *101*(1), (pp. 184–204).

Wang, Y., See, J., Oh, Y.-H., Phan, R. C.-W., Rahulamathavan, Y., Ling, H.-C., Li, X. (2017). Effective recognition of facial micro-expressions with video motion magnification. *Multimedia Tools and Applications*, *76*(20), (pp. 21665–21690).

Wang, I., Narayana, P., Smith, J., Draper, B., Beveridge, R., & Ruiz, J. (2018, March). EASEL: Easy Automatic Segmentation Event Labeler. *In 23rd International Conference on Intelligent User Interfaces* (pp. 595-599).

Wu, S.-Y., Thawonmas, R., & Chen, K.-T. (2011). Video summarization via crowdsourcing. In *Chi'11 Extended Abstracts on Human Factors in Computing Systems* (pp. 1531–1536).

Yuen, M.-C., King, I., & Leung, K.-S. (2011). A survey of crowdsourcing systems. In *2011 IEEE Third International Conference on Privacy, Security, Risk, and Trust, and 2011 IEEE Third International Conference on Social Computing* (pp. 766–773).