

Aalto University  
School of Science  
Degree Programme of Computer Science and Engineering

Samuel Suikkanen

# Gamification in video labeling

Master's Thesis  
Espoo, May 4, 2019

Supervisors: Professor Yu Xiao, Aalto University  
Instructor: Petr Byvshev M.Sc. (Tech.)

<b>Author:</b>	Samuel Suikkanen	
<b>Title:</b>	Gamification in video labeling	
<b>Date:</b>	May 4, 2019	<b>Pages:</b> 65
<b>Professorship:</b>	Networking Technology	<b>Code:</b> T-110
<b>Supervisors:</b>	Professor Yu Xiao	
<b>Instructor:</b>	Petr Byvshev M.Sc. (Tech.)	
<p>Machine learning is used in many fields in today's world. For many supervised and semi-supervised methods more accurate results can be achieved by gathering more labeled data. The increased amount of available data introduces problems to the labeling process. As a result, many crowdsourcing platforms have risen to help ease the labeling process. These crowdsourcing platforms offer monetary rewards for users that label data. However, the expenses start to be noticeable, as the databases grow larger.</p> <p>This thesis surveys the existing methods used in computer vision based data labeling, as well as designs and implements a game around video labeling using bounding boxes. Using gamification we aim to lower the cost of the labeling process.</p> <p>A user study was conducted to evaluate the game. User satisfaction and the labels created by the study participants were evaluated using a ground truth model. The user study showed us that building a simple game for the labeling process did not get the users engaged in the activity. The labels provided by the users were inaccurate. Area and standard deviation based filtering techniques were implemented to clear the data. These two filtering techniques helped to improve the accuracy of the labels. However, majority of the labels did not meet the accuracy criteria of 50% intersection over the union on average. The inaccurate labels are partly caused by the low density of answers. Thus, the accuracies are expected to improve with more answers.</p>		
<b>Keywords:</b>	Video labeling, labelling, machine learning, gamification, crowdsourcing	
<b>Language:</b>	English	

<b>Tekijä:</b>	Samuel Suikkanen		
<b>Työn nimi:</b>	Videoiden merkkauksen pelillistäminen		
<b>Päiväys:</b>	4. toukokuuta 2019	<b>Sivumäärä:</b>	65
<b>Professori:</b>	Tietoliikenneohjelmistot	<b>Koodi:</b>	T-110
<b>Valvojat:</b>	Professori Yu Xiao		
<b>Ohjaaja:</b>	Diplomi-insinööri Petr Byvshev		
<p>Koneoppiminen on laajasti käytössä nykypäivän teollisuudessa. Usein ohjatut sekä puoli-ohjatut menetelmät tarkentuvat, kun uusia syöte-tulos pareja saadaan kerättyä. Kuitenkin kasvavan datan määrä Internetissä aiheuttaa ongelmia syöte-tulos parien keräämisessä. Kasvun seurauksena monia joukkoistamis alustoja on syntynyt. Näitä alustoja käyttämällä voidaan palkata monia ihmisiä suorittamaan pieniä tehtäviä rahallista korvausta vastaan. Tietokantojen kasvaessa suuriksi rahalliset kustannukset kasvavat myös huomattaviksi.</p> <p>Tämä työ tarkastelee nykyisiä menetelmiä joita käytetään syöte-tulos parien muodostamisessa konenäön algoritmeille. Lisäksi työssä suunniteltiin ja toteutettiin peli, joka loi syöte-tulos pareja rajaavien laatikoiden muodossa. Pelillistämisen avulla pyrittiin leikkaamaan syöte-tulos parien luomisen kustannuksia.</p> <p>Pelin evaluoimiseksi suoritettiin käyttäjätutkimus. Käyttäjätuutuväisyyden lisäksi käyttäjätutkimuksesta saatujen syöte-tulos parien tarkkuutta arvioitiin totuusmallia käyttäen. Käyttäjätutkimus paljasti, että yksinkertainen pelimme ei saanut käyttäjiä kiehtoutumaan merkkauksesta. Käyttäjien vastauksista luodut syöte-tulos parit olivat epätarkkoja. Kahta suodatinta menetelmää käytettiin syöte-tulos parien puhdistamiseksi. Nämä menetelmät auttoivat parantamaan luodun syöte-tulos tietokannan tarkkuutta. Suurimmassa osassa tarkkuus ei kuitenkaan saavuttanut haluttua 50% keskiarvoista tarkkuutta leikkauksen ja unionin suhteessa. Osa-syy mallin epätarkkuuteen oli alhainen vastaustiheys. Voidaan olettaa, että malli tarkentuisi uusien vastauksien seurauksena.</p>			
<b>Asiasanat:</b>	Koneoppiminen, Videoiden merkkauk, joukkoistaminen, pelillistäminen, syöte-tulos pari		
<b>Kieli:</b>	Englanti		

# Acknowledgements

I would like to thank my supervisor, Yu Xiao for giving me the time and opportunity to write my thesis. Also, I want to address my gratitude to my advisor, Petr Byvshev, for his ideas and improvements he shared to me during the writing of this thesis.

Finally, I want to send my best regards to all the people who participated in the user study and to the people who listened my countless hours of desperate ramblings.

Espoo, May 4, 2019

Samuel Suikkanen

# Abbreviations and Acronyms

API	Application Programming Interface
CAD	Computer-Aided Design
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CNTK	Microsoft Cognitive Toolkit
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
MDA	Mechanics-Dynamics-Aesthetics; A framework tool used to analyze games
MVC	Model-View-Controller; Architectural pattern for developing user interfaces
R-CNN	Region-based convolutional neural network
REST	Representational State Transfer; Software architectural style for web services
SUS	System Usability Scale
URI	Uniform Resource Identifier

# Contents

<b>Abbreviations and Acronyms</b>	<b>4</b>
<b>1 Introduction</b>	<b>8</b>
1.1 Motivation: Activity recognition challenges . . . . .	10
1.2 Problem statement . . . . .	11
1.3 Structure of the Thesis . . . . .	12
<b>2 Background</b>	<b>13</b>
2.1 Crowdsourcing . . . . .	13
2.2 Gamification . . . . .	14
2.3 Label formats . . . . .	16
2.3.1 Natural language descriptor . . . . .	16
2.3.2 Bounding box and Polygon . . . . .	17
2.3.3 3D models . . . . .	17
2.3.4 Full-frame segmentation . . . . .	18
2.4 Games with a purpose . . . . .	19
2.4.1 Google image labeler and ESP game . . . . .	20
2.4.2 Peekaboom . . . . .	21
2.4.3 Other games . . . . .	22
2.5 Video labeling tools . . . . .	22
2.5.1 Video Annotation Tool from Irvine, California . . . . .	23
2.5.2 BeaverDam . . . . .	23
2.5.3 Computer vision annotation tool . . . . .	24
2.5.4 Microsoft VoTT . . . . .	24
2.5.5 Summary . . . . .	25
<b>3 Design and Implementation</b>	<b>26</b>
3.1 Game Design . . . . .	26
3.1.1 Requirements . . . . .	26
3.1.2 Proposed Games . . . . .	27
3.1.2.1 Determining correct answers . . . . .	30

3.2	Web app implementation . . . . .	31
3.2.1	Client . . . . .	32
3.2.1.1	Instructions and high scores . . . . .	32
3.2.1.2	Game view . . . . .	33
3.2.1.3	Database management . . . . .	35
3.2.1.4	Model preview . . . . .	36
3.2.2	Server . . . . .	36
3.2.2.1	REST API . . . . .	37
3.2.2.2	Database . . . . .	38
3.2.2.3	Segmentation . . . . .	38
<b>4</b>	<b>Evaluation</b>	<b>40</b>
4.1	User study . . . . .	40
4.1.1	Pilot test . . . . .	41
4.1.2	Users task . . . . .	41
4.1.3	Subject group . . . . .	42
4.1.4	Results . . . . .	42
4.2	Label accuracy . . . . .	44
4.2.1	Interpolation . . . . .	45
4.2.2	Post-processing filters . . . . .	45
4.2.3	First segment . . . . .	46
4.2.4	Second and third segment . . . . .	48
4.3	Summaries . . . . .	50
<b>5</b>	<b>Discussion</b>	<b>51</b>
5.1	Possible evaluation distortions . . . . .	51
5.2	Moving towards crowdsourcing platforms . . . . .	52
5.3	Improving the used methods . . . . .	52
5.4	Game design . . . . .	53
<b>6</b>	<b>Conclusions</b>	<b>55</b>
<b>A</b>	<b>Questionnaire</b>	<b>64</b>

# Chapter 1

## Introduction

Autonomous systems are evolving and making our life easier. These systems rely on different kind of sensors and computer vision is arguably the most important of them all, as human vision is a particularly rich source of information. Traditional computer vision task of detecting objects is not sufficient in many new applications. Modern robots need to recognize and predict activities, for example, automated driving systems need to predict crashes and prevent them. With accurate gesture recognition, programs could be made more interactive. [46] Controlling computers fluidly by hand gestures would make the classic scene of science fiction movies reality.

Different supervised and semi-supervised deep learning methods are commonly used for computer vision tasks. These algorithms are trained with an immense amount of labeled data. Usually, as more input is provided, more accurate results are achieved. Years of algorithm tuning can be compensated by just adding more labeled data [50]. Compared to the amount of data in the internet labeled data is a scarce resource [30].

Many datasets are available for machine learning [15, 21, 24, 36, 45, 59]. A small video dataset can include only six classes with 120 instances of videos [48]. The largest video dataset, YouTube-8m, consists of over 3000 classes with eight million instances of videos [2]. These databases can be used to train and test, for example, object detection, facial recognition, handwriting, and action recognition algorithms. Thus, these datasets can be used to reduce data collection and labeling costs. However, for specific purposes, like activity recognition, many of the databases are small in size and not sufficient for accurate learning. Especially for video labels, the databases are usually only labeled coarsely using natural language. For example, in the YouTube-8m database, a peel class includes long cooking videos where peeling is performed only in a short part of the video.

One of the reasons why existing video datasets are coarsely labeled is the

increased complexity of the labeling process. Fundamentally, a video consists of a stream of images, making the labeling process more time-consuming compared to image labeling. A video can consist of 60 frames per second, which quickly adds to thousands of images. In addition, videos can describe more complex scenarios and situations which can require several seconds or minutes of inspection for full understanding. These effects make the labeling process time-consuming. Yet, additional information can be gained from sequential frames. A video can be used to determine relationships, actions, movements and 3d models of objects. A video can be used to distinguish if a human is hammering or if the human is only holding the hammer in a hand. Using a single image this separation is impossible as images can not represent movement. Manual work is required to label this information in the videos.

Hiring labeling experts to label data is expensive. Crowdsourcing is a common way used to cut the costs of the labeling process. Video and image labeling does not require any special expertise, making it a great candidate for crowdsourcing. A batch of images is easy to break down to small subproblems as every image can be labeled independently. Using the same approach for videos is difficult, as videos can vary in length. Thus, additional considerations must be taken into account on how to break down the task. Only relevant parts of the video should be displayed to the user to achieve high efficiency.

Many crowdsourcing platforms offer monetary rewards to users who are willing to perform small tasks. When large datasets require labeling, the monetary crowdsourcing starts to build up costs. We address this problem by examining different incentive mechanisms. Could other incentive mechanisms make the process more efficient? Could we get people to perform labeling without the monetary incentive?

One method used today to cut costs is the completely automated public Turing test to tell computers and humans apart (*CAPTCHA*). The test is used to determine real users from malicious bots in websites. The users are not allowed to use the website without performing the task. Commonly users are asked to indicate locations of objects in images by clicking on sections which include parts of the object. Multiple different versions of CAPTCHA are available and the tests are designed in a way that they output labeled data for deep learning purposes. [35]

The gaming industry has grown to be a leading force in today's market [38]. The rise of smartphones has made games available to every citizen. By implementing a popular game around the labeling process, we could label millions of videos in an efficient and cheap manner. Some successful games have been built around the labeling process [37, 53, 54]. For computer vi-

sion, these games have focused on images or specific scientific topics, like brain mappings.

This study reviews the current methods used in the labeling process and tries to find new incentive mechanisms for video labeling through gamification. We did not find any games designed for video labeling. We implemented a prototype game around the labeling process and conducted a user study to evaluate the game. We recruited 20 users to test the game and to answer a questionnaire about the game. In addition, we built a ground truth model to analyze the accuracy of the user provided answers.

## Motivation: Activity recognition challenges

Activity recognition has importance across many different research areas including human-computer interaction, rehabilitation engineering, human-robot interaction, assistive technology, and autonomous driving. Two classes for human activity exist, simple human activity and complex human activity. The first class involves body motion and posture, such as running, walking and sitting. The second class includes more complex scenarios and interactions with objects, such as cooking, reading and watching tv. [40]

Building an autonomous system to recognize and predict activities is challenging because activities are complex and highly diverse [31]. Recognizing simple human activities in controlled environment can be relatively easy for computers. In real-life applications recognizing actions comes increasingly difficult due to different camera angles, video quality, and intra-class variation. [33]

Building 3d models from videos is expensive and difficult. For this reason, action is commonly represented in a holistic or local way. Herath et al. [28] define these two as follows:

- Holistic representations. Action recognition is based on the extraction of a global representation of human body structure, shape and movements.
- Local representations. Action recognition is based on the extraction of local features.

Intra- and inter-class variation is the variation within a class and between classes. Intra-class variation is caused by one action category containing multiple different styles of human movement. The same activities can be performed differently, thus they have a high intra-class variation. As a result, different postures, trajectories, and camera angles can affect the

recognition [45]. High variation within the class makes recognition harder as it is hard to generalize the features for the class. Thus, increasing variation within the class requires more training data. [43]

Inter-class variation is the variation between classes and it depends highly on the label space. For simple binary classification, determining if the person is lying or standing, the variation is big. In contrast, recognizing different cooking activities is difficult as they have small inter-class variation. Thus, high inter-class variation makes the recognition easier. The goal of the recognition algorithms is to learn the distinct features differentiating the activities, i.e., to select features which minimize the intra-class variation while maximizing the inter-class variation [43].

When training action classifier, the videos should only contain the relevant information associated with the class. As stated previously, in the YouTube-8m, a peel class can contain multiple different cooking activities in one video. Learning the features which maximize the inter-class variation of different cooking activities becomes challenging. Thus, the activity classes should accurately define the relevant time segments for the class.

Motion representation is an important part of classifiers using holistic features. However, when background noise is present, it is difficult for the algorithm to distinguish object motion from the background. Bounding box labeling can be used to reduce the effects of background noise. By using bounding boxes, the separation of the background is already done by the labeling process. [33] Many of the existing databases lack these kinds of low-level labels. We believe that activity recognition classifiers can be improved with more appropriately labeled datasets.

## Problem statement

Computer vision labeling is easy to crowdsource, as the task can be broken down to smaller parts and the results can be aggregated easily. In addition, the labeling process does not require any special expertise. In recent years many specialized crowdsourcing platforms have risen. These crowdsourcing platforms offer monetary incentive to recruit workers for tasks.

This thesis tries to answer the following research question: *How can we get the users to enjoy this monotonous and repetitive task?* Thus, the core problem of the thesis is the masking of a monotonous and repetitive task in a clever way. Using gamification we hope to reduce the costs of the labeling process. Secondly, this thesis examines the current methods and formats used to label video data.

We propose a gamification method for introducing intrinsic motivation

to annotators. When the work is monotonous and repetitive, the intrinsic motivation can help the users to be more focused on the task in hand. In addition, we were motivated by the immediate effect of reducing the costs, as people could be playing the game solely for the purpose of having fun. We built a game around the labeling process and conducted a user study to evaluate the system.

## Structure of the Thesis

This thesis is structured as follows. Chapter 2 researches the relevant background for the thesis. Chapter 3 covers the requirements for the game design and discussed the implemented game in detail. The chapter covers the implementation of the server and the client, explaining the data structures and the algorithms used in the game. Chapter 4 explains the evaluation methods and analyzes the results. A user study was conducted with 20 subjects and user satisfaction and the accuracy of the outputs were evaluated. Chapter 5 discusses and suggests possible areas for future research. Lastly, the chapter 6 gives a brief overview of the whole thesis.

## Chapter 2

# Background

Video labels can be anything from a keyword defining the general topic of a video to fine-grained separation of objects and their relations frame by frame. The annotation level issues from the intended context of usage. For example, a keyword can be used in video retrieval to match user query with relevant videos. These kind of coarse labels are rarely sufficient for deep learning purposes. Generally, for deep learning purposes accurate low-level labels are useful and provide good results. In addition, it is easier to convert fine-grained data to coarse than vice-versa. [14]

In this chapter we discuss crowdsourcing and gamification, which are commonly used in data mining tasks, such as labeling of computer vision data. In addition, we survey the relevant tools and data types used in computer vision labeling.

## Crowdsourcing

Crowdsourcing is a sourcing model in which work is divided between participants and results are achieved cumulatively. Crowdsourcing has played a big role in data mining, a term used to describe the process of extracting information from data set and convert it into a comprehensible structure for future use. [25] Crowdsourcing can be used in many different environments. For example, current navigation systems, like Waze, use crowdsourcing to gather information about road conditions, traffic jams, and police raids. Users can report sightings to the software which are then collected and displayed to the other users. [9]

In recent years many specialized crowdsourcing platforms have risen. Amazon Mechanical Turk is one of the most popular ones, allowing users to give tasks for a large group of people. [25] Tasks are attached to a mon-

etary reward. The monetary reward works as an incentive for the workers. Usually, the tasks are small and the rewards paid for the task are between two to ten cents. Video labeling task is easy to break down to small independent parts and the results can be then aggregated. This makes it a good candidate for these monetary crowdsourcing platforms. [16]

A monetary reward is a form of extrinsic motivation. Human motivation can be generally categorized by intrinsic or extrinsic motivation. People operate by extrinsic motivation when they are pursuing a reward or trying to avoid punishment. [47] Monetary incentive attracts malicious users. People want to abuse the systems to gain profit. Thus, a good verification method is required in crowdsourcing platforms. [19]

A popular labeling method using extrinsic incentive mechanism is CAPTCHA. CAPTCHA is used in many websites to separate malicious bots from humans. The users have to perform a small task to access a website, for example, indicate locations of objects in images by clicking on sections which include parts of the object. The tasks are designed in a way that the output from the users can be used as labels for data. The users are willing to perform the task because they have the incentive to use the website. [35]

Intrinsic motivation is defined as the doing of an activity for the fun or challenge entailed, rather than the desire for some external reward. Games are typically played because of intrinsic motivation, people have fun while playing games. Gamification is one way used to introduce intrinsic motivation to the users and is discussed in more detail in the next section. [47, 54]

## Gamification

People have enthusiasm and engagement for their hobbies, they can spend hours on the things that give them enjoyment. Gamification tries to bring this enthusiasm to everyday work. Gamification is described as either using game elements in a non-game environment or making the systems more intrinsically motivating through structural or organizational changes. Crowdsourcing is one of the largest domains for gamification [32]. Many platforms offer crowdsourcing using extrinsic, monetary compensation. However, the monetary incentive does not make the user engaged in the activity. While the labeling process can be easily broken down to small simple tasks, the process is rather monotonous and repetitive which makes workers prone to errors and sloppy mistakes. [52] This can be further seen at the research by Vondrick et al (2012) which states that the average worker at Amazon's Mechanical Turk does a poor job at labeling [56].

Contrary to the extrinsic incentive, users can feel much more engaged in

tasks when they are driven by intrinsic motivation. Gamification is one way to introduce intrinsic incentive to the process, as games are considered an effective way of fulfilling intrinsic needs. Users need to feel competent, have a sense of autonomy, be challenged, and have competition and cooperation among other players. Especially features that invoke competition have been found to improve the performance in monotonous and repetitive tasks. [41]

A game can be analyzed by its mechanisms, dynamics, and aesthetics. This is commonly referred as the MDA framework [29]. The game mechanisms define the rules and every basic action that the user can take. Dynamics are the behaviour of the mechanics acting on player input. Aesthetics are defined as the emotional responses resulting from the gameplay. [61] The aesthetics can be categorized into eight types: sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission [29].

Common game mechanisms used in gamification include levels, points, and leader boards. Leader boards are used to promote competition and have been found to improve performance in monotonous and repetitive tasks [41]. Players can be informed about which activities are more important by punishing or rewarding them with points. Redeemable points can be used to create virtual economies inside games. To create a successful credit system, users need to be able to redeem the points for something valuable for them. A well-designed credit system is a valuable asset for building large communities inside games. [61]

Levels are commonly used to indicating the progress of a game. Traditionally, games become more challenging as the levels progress. The increasing challenge creates a feeling of satisfaction, the users feel competent and have a feeling of mastering something. Achieving this kind of progressive difficulty is a challenge in many gamification contexts. For example, the video labeling game should produce accurate labels for objects. However, designing a game with progressive difficulty while maintaining the accuracy of labels is challenging. Levels can be also displayed using progress bars and achievement badges. A study by Hakulinen (2014) shows that students were spending more time in the online learning environment when achievement badges were used in a university course [26]. The study also summaries that students with high motivation were the most affected by the badges. Thus, it is important to note that using leaderboards and badges does not magically make people enthusiastically work on the task at hand.

Games need to be thought out carefully to make them entertaining [61]. The goal of a game is to create emotional responses in the players. Leader boards and achievement badges can be used to create competition. Increasing difficulty can be used to create challenges to the users. A chat and multi-player mode can be used to promote fellowship to the users. [29] A game can

implement multiple different aesthetic components to motivate users. The game designer is responsible for designing the mechanisms and the dynamics of the game to create the aesthetics of the game. It is not always necessary to make a complex game, a simple interaction with other people can make already motivated worker more engaged [52].

## Label formats

There are multiple ways in which a dataset can be labeled. The used format should be decided based on the context of the usage. Different formats are commonly coupled to provide more accurate information about the data. This chapter discusses the different formats used when vision data is labeled. We start from simple natural language models to more fine-grained descriptions. Table 2.1 summarizes the advantages, disadvantages and common usages of the different labeling formats.

### Natural language descriptor

Natural language descriptor is the most intuitive way of labeling videos, explaining what is happening in them. A simple method used in the EPIC-KITCHEN dataset is commentator [13]. The user performing actions commentates their intentions, which can then be extracted to a file with appropriate timestamps. In the construction of the dataset the commentators were asked to describe their actions in simple sentences and crowdsourcing was used to verify the start and end times of the labels. Common labels in the database include the verbs put, take, wash, and, open with corresponding nouns. [13]

Furthermore, a simple title or a keyword can be thought of as a natural language descriptor. The YouTube videos are a good example of these kinds of annotations as the uploaders provide titles and keywords describing their videos. The largest labeled video dataset, YouTube-8M, leveraged this information during its construction [2]. The initial descriptions were inaccurate due to advertisement tactics employed by the uploaders, incorrect keywords are inserted in hopes for more views. The descriptions were filtered using different post-processing methods.

The downside of natural language descriptors is the lack of positional information. Natural language can be only associated with specific points of time in a video. Therefore, relevant areas of interest in the video frames can not be specified. The following chapter explains the polygonal and bounding box annotations, which are common ways of displaying areas of interest.

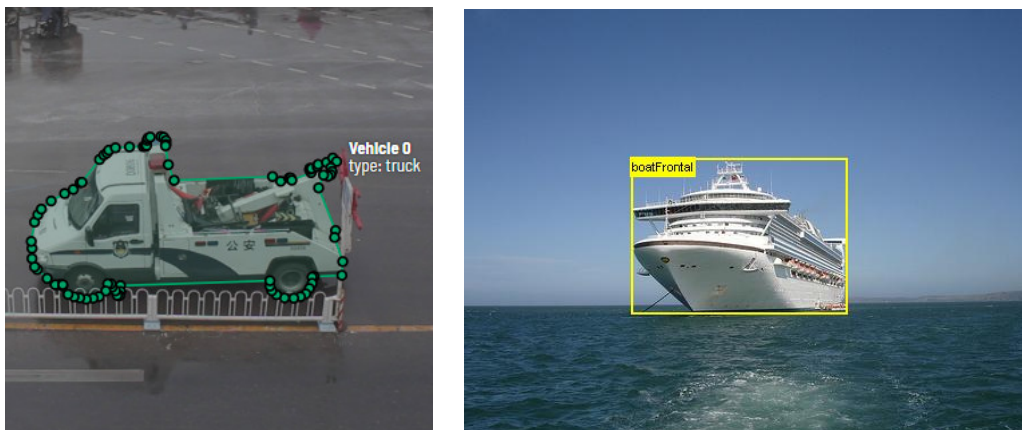


Figure 2.1: Left: Polygon annotation. Right: Bounding box annotation [21].

## Bounding box and Polygon

A bounding box is a simple way of displaying the section of interest by providing coordinates for a rectangle in a image. This benefits the learning model greatly, as it is easier to extract important features when the section of interest is known. In addition, the effect of background noise is minimized [33].

For videos, the data is usually saved in a separate file. The bounding boxes are given for specific frames, and interpolation can be used to calculate the bounding boxes location between the frames. The bounding box is usually described in corner point coordinates or with the center point combined with width and height.

Polygon annotation is described by a finite number of points, making more accurate cropping of objects possible. In the figure 2.1 the difference between polygon and bounding box annotation are shown. Polygon annotations are usually not available in public datasets. This is probably because creating them is slow compared to the achieved accuracy increase in the learning model.

## 3D models

Perceiving objects in depth is a fundamental ability of the human vision, which is something that modern robots try to achieve. Most flexible tasks require 3D spatial awareness. Automated systems need to detect poses, locations and velocities of objects to work reliably in complicated environments. [18] Additional sensors can be used to display the spatial structure of an image. Most commonly, RGB-D sensors are used to include depth

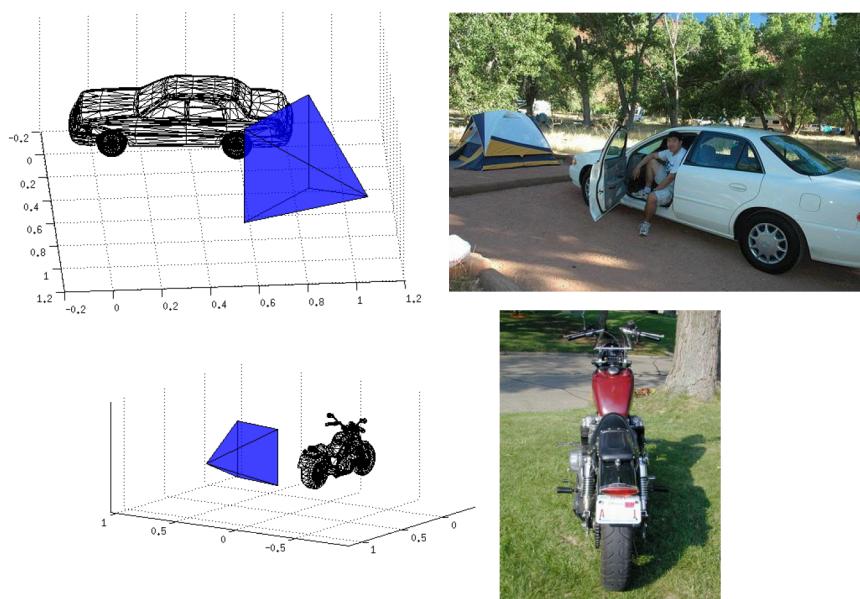


Figure 2.2: An example from PASCAL3D Dataset. The most suitable 3D CAD model is chosen from existing models, and then the model is aligned with the image [58].

information to videos.

The labeling is usually done by using a computer-aided design (*CAD*) program. These programs can be used to specify camera locations and point clouds which are linked to the images or frames. An example of the method is shown in figure 2.2. This is a rather time taking process, and some datasets just label the photos with estimated 3d bounding boxes. [23, 39].

## Full-frame segmentation

Full frame segmentation is a term used when all the areas of the frame are labeled, every pixel is classified to some label. In reverse, a dense projection is a method of trying to predict a label for every pixel in a picture.

For some applications, it is important that the whole frame is perceived thoroughly. For example, self-driving cars have extreme demands on system reliability and performance. Therefore, it is important for these cars to understand complex traffic scenarios and driving scenarios. When a person is standing next to a road, it does not necessarily mean that he is going to go on the road. A car is not expected to stop if the person is standing next to a highway. In contrary, if the person is standing next to a pedestrian crossing the car is expected to stop and wait for the person to pass the road. Thus,

it is vital for self-driving cars to understand the context of the situation.

In addition, full-frame segmentation has been shown to bolster dense prediction and object detection [60]. Many of the existing databases [11, 21, 59, 60] have full frame segmentation annotations.

	Advantages	Disadvantages	Common in
Natural language descriptor	Intuitive, Fast labeling process	No location information	Most datasets
Bounding box	Defines areas of interest, Reduces background noise	Slow labeling process	Object detection
3D models	Spatial awareness	Limited tools, Complex labeling process, Requires CAD experience	Modern robot interaction, specific cases (Brain mappings etc.)
Full-frame segmentation	Fine-grained labeling, Can bolster object detection and activity recognition	Requires more manual work than bounding box labeling	Autonomous driving and scene detection

Table 2.1: Summary of the different labeling formats

## Games with a purpose

Games with a purpose (*GWAP*) is a term used to describe games that are used with an underlying purpose, for example, labeling data or achieving education for students. Many of the successful GWAPs that are used in data labeling can be categorized into three distinct types [54].

- **Output agreement games:** Players are given the same input and must agree on the output.
- **Input agreement games:** Players are given different or same input and they have to decide which of the states holds true.
- **Inversion-problem games:** Player can be the problem describer or guesser. Based on the describers hints, guesser has to find the correct answer.

## Google image labeler and ESP game

In 2006 Google launched an imaged based game, with the underlying purpose of verifying the labels used in their image search. The game was originally released under the name ESP game [54]. Google bought a license to build its own version of the game. The game was played with random pairs, and if an uneven number of players were online, a player could play with a predefined dictionary.

Pairs try to achieve group consensus on the label for a picture. If both players guessed the same label, points were rewarded. General labels gave fewer points than accurate labels. For example, identifying a person by her or his name gave more points than just stating that the image includes a person. On the newly launched version, two players try to agree which picture they share on two displayed pictures. The players are allowed to ask questions from each other. The fewer questions were used to achieve group consensus, the more points were rewarded.

Users can create profiles and leaderboards were displayed for the best players and teams. Google relied on the competitiveness of the users to keep on playing, and these statistics were used as the temptation. Similarly to VATIC in Chapter 2.5.1, malicious users were encountered. For example, predefined dictionaries to brute-force the guesses. [54]



Figure 2.3: The ESP Game [54]

## Peekaboom

The ESP game could only provide coarse-grained labels for images, the labeled images did not have any metadata, information about the locations of the interesting objects in the images. As a successor of ESP game, Peekaboom tries to address this problem. By taking the images with labels from the ESP game and using them in Peekaboom, bounding boxes can be determined for the labels. [55]

The game follows a pattern similar to the ESP game. Pairs of users are given a task. In the game, players take turns being the peek or the boom. The boom is given an image and a word related to the image. The goal is that the boom reveals parts of the picture in a way that the peek can come up with the related word. The game also has a bonus round which is played after four completed images. In the bonus round, both users are shown a picture and asked to click on a related word. The distance between the clicks determine how many points they get. [55]

The bounding box for an object is generated in a simple way. First, a matrix is made of all the revealed sections. Then all sections that have been revealed only once are discarded. The remaining areas are used to determine the bounding box.

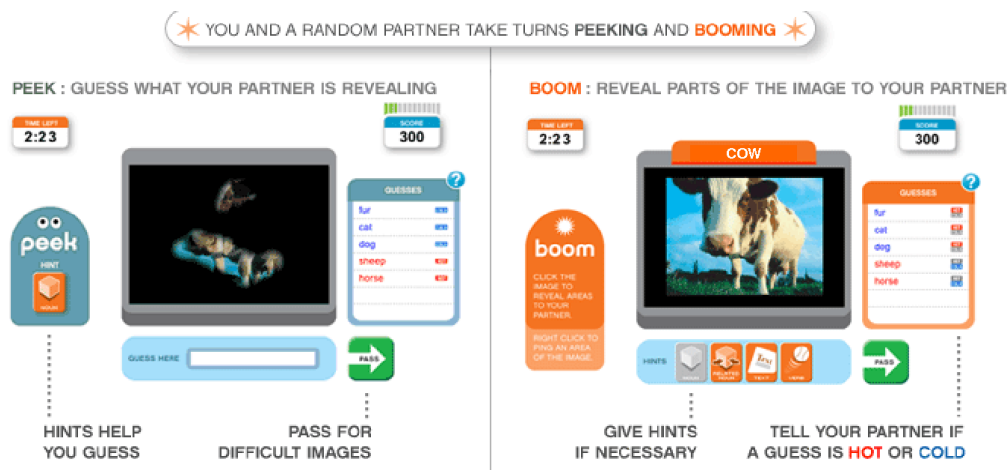


Figure 2.4: The Peekaboom game [55]

## Other games

There are many different kinds of GWAPs available and not all of them are related to computer vision. In a popular game called EyeWire <sup>1</sup>, users are asked to identify connected areas in functional magnetic resonance images (*fMRI*) to help neuroscientist understand how the brain operates.

Biochemistry has also released multiple games with a purpose. Games like Phylo, Foldit and EteRNA are disguised as puzzle games [10, 34]. By playing the game users help researchers with complex tasks. In EteRNA users design sequence that can fold into RNA structures. The design is done at the interface displayed in figure 2.5. By clicking on the nucleotides, represented by the colored circles, users can change the connections in the RNA structure. Users vote for the best designs and the eight top-voted sequences are synthesized and verified by the authors. [37]



Figure 2.5: EteRNA design interfaces

## Video labeling tools

Deep learning is not the only appliance for labeled data. For example, there are multiple tools available for annotating videos so that they can be leveraged by video retrieval methods. These tools usually label the data in more general level and use existing descriptions for the labeling process. Therefore, these tools are not as useful for machine learning purposes and are not surveyed in this section. [14]

---

<sup>1</sup><https://eyewire.org/explore> [Accessed 11.04.2019]

We did not find any games with the purpose of labeling videos. However, many tools are available. This section gives a brief introduction to the most popular tools. The most common way of labeling videos in these tools is bounding boxes. The list is not supposed to be a comprehensive view of all the available tools. Rather in this study, we try to outline different concepts and ideas in the tools that are used to achieve efficient labeling.

## Video Annotation Tool from Irvine, California

Video annotation tool from Irvine, California (*VATIC*) attempts to lower the costs of manual labeling by using crowdsourcing [56]. The application is not well maintained and is considered an ancestor for many newer applications. The program was originally developed to be used with Amazons Mechanical Turk. Videos are broken down to images frame by frame. Then small segments of the images can be processed in parallel and results are merged progressively. Annotations are done every keyframe and interpolation is used to move the bounding boxes.

Vondrick et al (2012) recognized that the majority of the workers are either incompetent of labeling videos or trying to cheat to gain profit [56]. Verification in many of the existing tools is implemented by asking the same question from multiple workers and then comparing the answers with some accuracy method. The correct answer can be then determined by selecting the answer with the highest accuracy score. *VATIC* approaches the verification differently. In *VATIC* every user is first hiddenly forwarded to a special task which is used to determine if the worker is competent. The task is used to filter incompetent and cheating users, if the worker does not pass the first task they are not allowed to annotate. Workers who pass the hidden task are trusted and allowed to work on the annotation problems independently. Thus, no additional verification is used when a trusted user annotates a video segment. [56]

## BeaverDam

BeaverDam is a frame-by-frame bounding box annotation tool [50]. The tool tries to improve the usability and the efficiency of *VATIC*. One disadvantage of *VATIC* is the lack of batch controlling, which BeaverDam addresses by implementing a graphical interface for management. In addition, the tool uses little dependencies which makes it easy to set up. The code is written modularly, making it configurable and versatile for specific use cases.

One big difference in BeaverDam is that it uses the HTML5 video element on the client side labeling. This means that the video is not broken down

to pictures on the server side. This makes the load on the server lighter. However, the author has recognized that when using the native video format, accuracy can be affected. [50]

## Computer vision annotation tool

Computer vision annotation tool (*CVAT*) is a web-based tool inspired by VATIC [6]. The tool uses Docker, a popular container software for the operating-system-level virtualization [17]. Docker makes launching and scaling simple.

CVAT provides attributes for labels. The attributes can be used to add additional information for the labels. For example, the user could add information about objects condition, color, age, and posture. The annotation interface is easy to use and responsive. The image can be moved and zoomed easily. To reduce manual work, TensorFlow can be used to predict labels in the videos.

The application relies on in-house labeling. Labels are not verified and there is no support for Amazons MTurk. Two users can not work on the same task concurrently. This can introduce scaling problems if the tasks are too large. The possible scaling problem can be mitigated by creating small tasks that are designed to single individuals.

## Microsoft VoTT

Microsoft's Visual Object Tagging Tool is an Electron-based application for generating and validating image and video labels [12]. A renewed version of the application was released during the writing of this thesis. Projects can be converted from the first version to the renewed version.

The first version of the tool is designed to be used in a single computer. Files are read and written from local storage. Labels can be predicted using region-based convolutional neural network (*R-CNN*) and labels can be tracked using camshift algorithm. The tool can also be used to validate Microsoft Cognitive Toolkit (*CNTK*) results.

The renewed version of the tool focuses on project management and project sharing. Projects are configured by creating connections which define the source and target of the project. Source connection defines the location of the assets that require labeling. Target connection defines the location to which labels should be exported. The source and target connections support Azure Blob Storage, thus, the projects can be shared and worked in collaboration with other people.

The second version of the tool has dropped the support of label prediction and label tracking. Instead, the new version focuses more on the labeling interface. The layout is more intuitive and additional hotkeys for label manipulation were added.

## Summary

All of the tools seem to provide a way for a user to define labels and associated bounding boxes with them. The location and size of the bounding box are decided by the user. The interfaces enable previewing and traverse of the video, enabling the users to find correct points in time for the bounding boxes. The tools usually use interpolation between labeled frames, so the user does not have to label every intermediate frame.

Almost all of the existing tools are used from web-browser and most of the tools can share the workload to multiple computers. Many of the tools have limitations on work sharing. For example, many of the tools can not handle concurrent workers on the same video segment. The tools that focus on concurrency are focused heavily on Amazon MTurk. The annotations are usually built by only accepting one worker's answers for a video segment [13, 27, 56]. The concurrent answers are compared with each other and the most suitable one is selected by using some accuracy measurement. Thus, the majority of the answers are not used to build the annotations.

## Chapter 3

# Design and Implementation

This chapter discusses the requirements for the game and the chosen architecture for the game implementation. Section 3.1 presents the considerations for the game design as well as the different proposed games. Section 3.2 discusses the implementation in detail.

## Game Design

This chapter discusses the requirements of the game and how those affected the game design.

## Requirements

The goal of the game is to generate labeled imagery data. The following main points were considered when making the decision:

- **Context of usage:** The output should be useful for activity recognition task.
- **Gamification:** The process of labeling should be constructed in an engaging way through gamification.
- **Label space:** The game should produce labels which are useful in machine learning.
- **Accessibility:** The game should be easily accessible to new users.

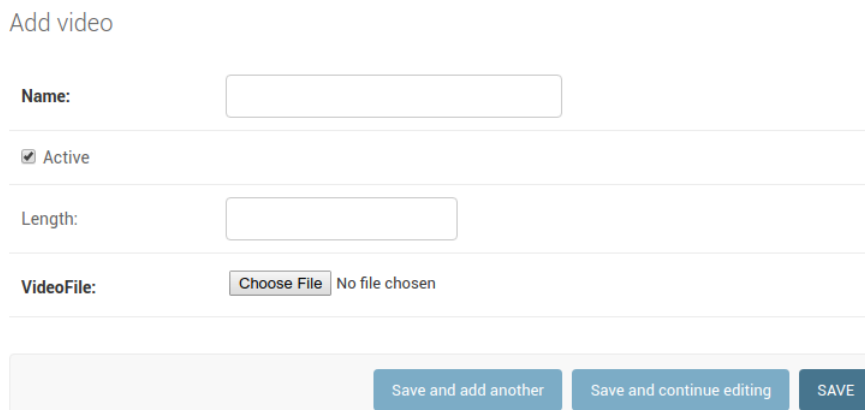
No standardized data format for video labels seems to be available. Many different formats are available which describe the same information. This makes the conversion between the formats straightforward to implement.

## Proposed Games

The underlying purpose of gathering labeled data constraints the game design. Three different approaches were implemented and all of the games were designed to give a different kind of output:

- **Label vote game:** Users were asked to give subject - verb - object definition to the video segment.
- **Click game:** Users were asked to click on objects.
- **Bounding box game:** Users were asked to draw bounding boxes around objects.

The label vote game was designed to give natural language descriptors for videos in a form of subject - verb - object. The game was designed to address the two consistent problems in existing datasets. Firstly, labels in existing datasets have coarse-grained labels defining general categories. We addressed this problem by implementing a structured format to categorize actions in videos. Secondly, the natural language descriptors usually do not specify points of times in the relevant sections of a video. This problem was addressed by letting the users specify the start and end point of the relevant part of the video.



Add video

Name:

Active

Length:

VideoFile:  No file chosen

Figure 3.1: Add video page

In all of the games videos are added through the administrator page. The page is displayed in the figure 3.1. The server is responsible for calculating the lengths of the videos. The videos are segmented to ten-second clips. These clips were displayed to users. Users were asked to give labels to these

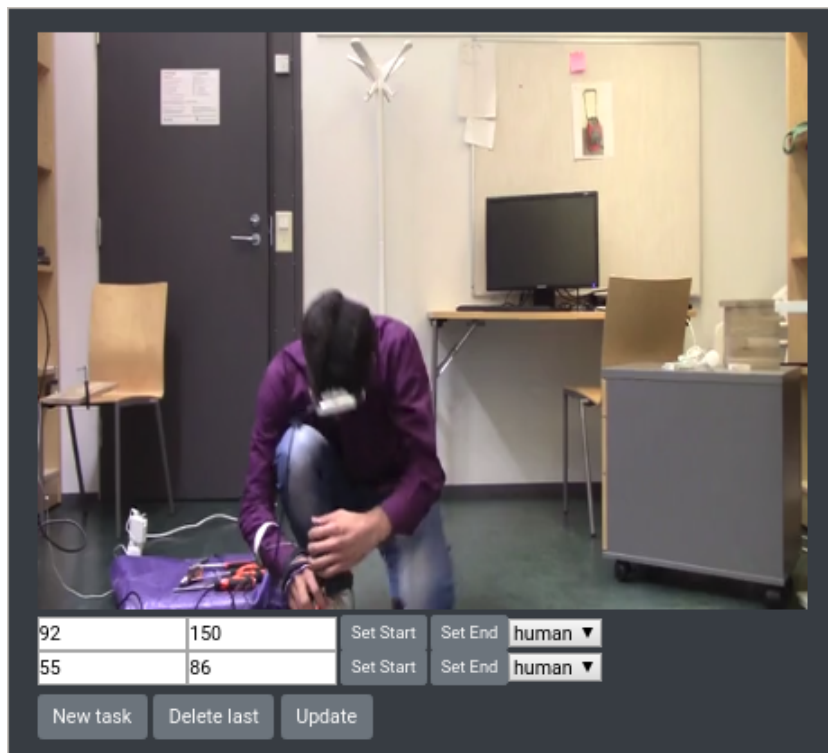


Figure 3.2: Manage tasks page

ten-second clips. Users could also extend or trim the video length so that the labels would have the correct start and end points in time. There were no restrictions on the label space, users could suggest any label. However, when a user started typing previously given subjects, verbs, and objects were suggested to the user. In the second stage, five different users were asked to vote for a label and majority rule was used to determine if the label should be accepted or rejected. Users agreeing with the majority and defining correct labels were rewarded with points, in contrary, users who were voting incorrectly or giving false labels were given penalty points.

We recognized that typing labels for videos do not make the game engaging, the game felt more like a tool for generating labels. The users were given too much freedom, the label space was infinite. This game was quickly discarded and the click and the bounding box game was developed instead.

The bounding box and click game were designed to give bounding box annotations for videos. Both of the games work in a similar fashion. In addition to videos, administrator users had to define questions and tasks. A task is defined by its video, question, start and end point of time, i.e.,

they specify which video segments require labeling and which objects in the segment require labeling. A task management page was implemented and displayed in figure 3.2. The page can be used to create and remove tasks while previewing video.

When a user starts to play the game a random task is selected and within the task, a random ten-second clip is rendered to the user. The question defined by the task is displayed to the user. In the bounding box game, the user answers the question by drawing a bounding box around the object displayed at the top of the video. In the click game, the user is shown the same information, but the answer mechanism is different. Instead of drawing a bounding box the user is asked to click on the objects in the click game. The only difference between the bounding box and the click game is the answer method for the questions. In the end, we chose to use the bounding box game, it feels more interactive and the answer mechanism is more helpful when building the output model.

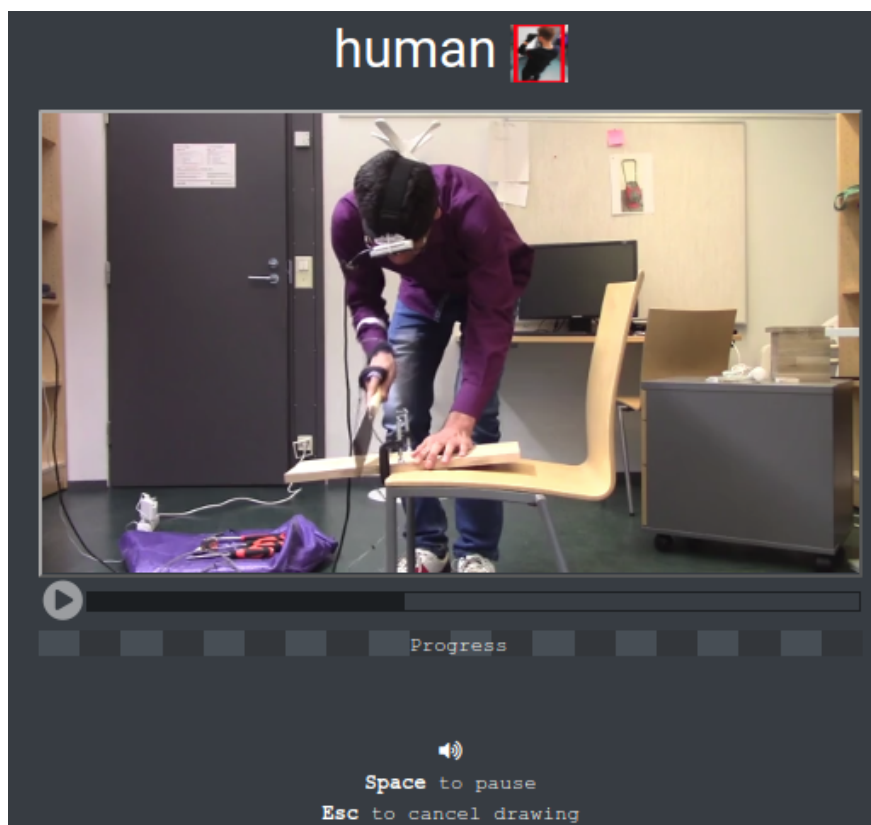


Figure 3.3: Gameplay

### Determining correct answers

Group consensus was used to determine if the bounding box given by the user was correct or incorrect. Intersection over union (*IoU*) was used as an accuracy method. In IoU, the overlapping area of  $A$  and  $B$  is divided by their area of union. The equation 3.1 shows how IoU is calculated for areas  $A$  and  $B$ .

$$IoU = \frac{A \cap B}{A \cup B} \quad (3.1)$$

The bounding box given by the user was compared to previously given bounding boxes at the nearby frames. A time window of one second was used. The answer is considered correct if it scores over 65% IoU for the majority of the nearby answers. An example algorithm of the verification process is written below.

```
def correct(answer):
    answers = get_nearby_answers(answer)
    correct, incorrect
    for a in answers:
        iou = get_iou(answer, a)
        if iou > 0.65: # Thershold for IoU
            correct++
        else:
            incorrect++
    return correct/(correct+incorrect) >= 0.5
```

The setback of this algorithm is the trade-off it introduces. When the time-frame gets larger in size, fast moving objects were usually considered incorrectly answered. Secondly, if there are no previous answers, it is impossible to determine if the user was correct or incorrect. Increasing the time frame increases the probability of gathering previous answers. However, increasing the time frame results in lower accuracy with moving objects.

The fixed values used in the study were determined to provide acceptable results. At the end of the user study, we started getting reports of the game providing an unusual amount of incorrect responses. This was due to fast moving objects analyzed in a too long time frame, thus a more dynamic approach that estimates the trade-off would be more preferable.

## Web app implementation

There are many options to consider when deciding the platform of the system. Labeling tasks are traditionally performed on a computer and that is why we decided to create the game to a website. A website makes the game easy to access and focuses on computer usage. In addition, browser-based applications can be used on almost any platform, as modern smartphones ship with a web browser. Creating native applications for all phone platforms would be a huge task. [8]

We chose to use the Django framework <sup>1</sup> as the basis for the project. Django has a large community behind it, which makes it easy to find answers to common problems and bugs. In addition, Django projects are created in a predefined structure which makes them easy to program against.

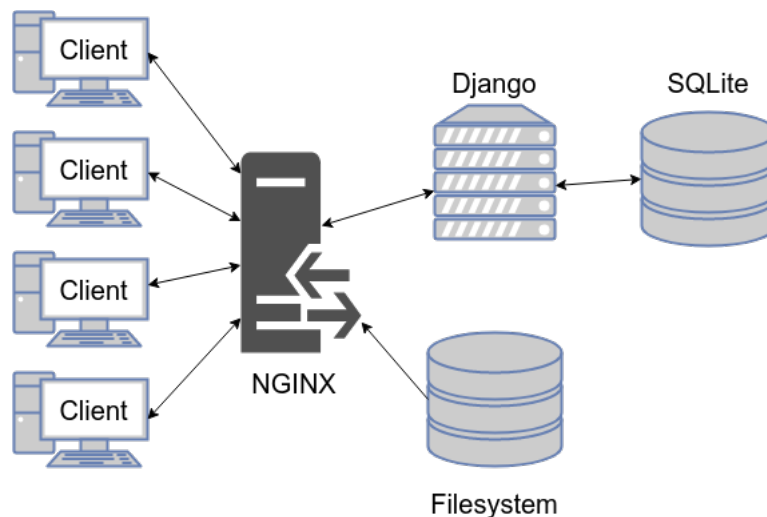


Figure 3.4: The clients communicate with the back-end through the reverse proxy. Static files are served from the filesystem.

The figure 3.4 displays the relationships of different parts of the architecture. The clients communicate with nginx proxy server with HTTP requests. Nginx works as a reverse proxy, forwarding the appropriate messages to Django and serving the static files straight from the file system. The Django service communicates with the users through nginx and uses the SQLite database to store relevant information.

This section describes the server and client implementation in detail. Most of the calculations and modifications for the data are done on the

<sup>1</sup><https://www.djangoproject.com/> [Accessed 30.4.2019]

server side. This was done for two reasons: the server helps the client to be as lightweight as possible while providing access control.

## Client

The client is designed in a way that the different modules are reusable and they can be combined effortlessly. For example, the navigation bar is created in a separate template. Django framework enables nested HTML templates and this navigation view is included in every view on the web page. The templates can be nested using Django's template syntax. The back-end is responsible for rendering the templates.

Today websites have evolved from static pages to more dynamic and data-driven sites where information is rendered to users asynchronously. Thus, to make the user experience more fluid we took advantage of the JavaScript capability of manipulating the HTML Document Object Model (*DOM*). Combined with AJAX calls to the server the page can be updated with necessary data without loading the page. [51]

Three main views were implemented to the client. These views were for the game, database handling and previewing the annotations. The client was implemented using HTML5, CSS, and JavaScript. Many of the components were created by the JavaScript dynamically to make the user experience more fluid. Additionally, few extra libraries were used to create the client, most notably the component library Bootstrap and the JavaScript library jQuery. JQuery was used to communicate with the server REST API. The library has easy to use methods for sending the requests and handling the responses from the server. Bootstrap was used in combination with HTML and CSS to style the page.

The instructions page is first displayed when a user comes to the website. The page explains the rules of the game and allows the user to enter the game page. A login button is displayed in the navigation bar, which allows users to register and login to the website. Logged users have the ability to preview the annotations built from the game. In addition, administrator users have access to manage tasks page and the Django admin dashboard.

## Instructions and high scores

The welcoming page displays instructions and the high scores for the user. During the development process user feedback of unclear instructions were gathered and as result pictures of example annotations were displayed. The home page can be seen in figure 3.5.

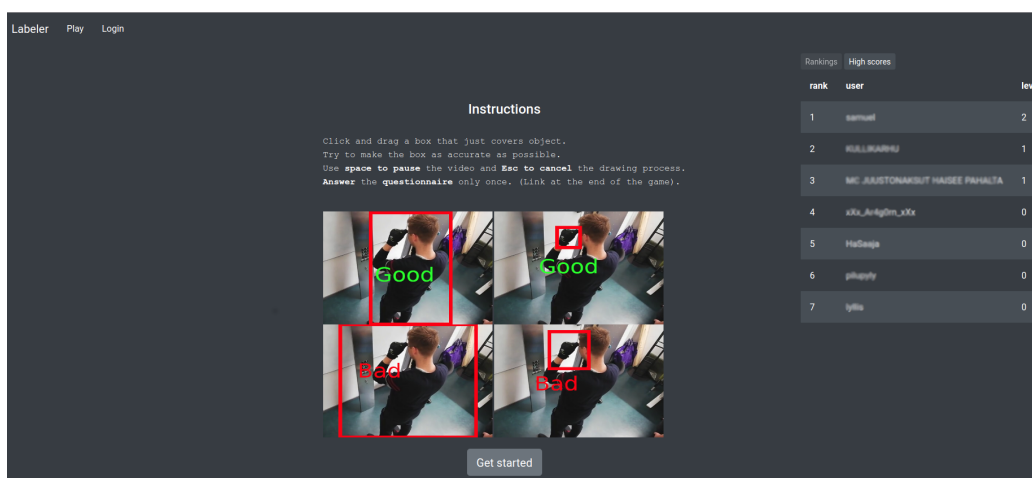


Figure 3.5: Home page with the high scores list on the right.

The high scores list is displayed on the right side of the page. The scoreboard is generated inside a JavaScript file. Only a placeholder tag is needed on the HTML page with the JavaScript added as a script tag to the HTML page. The scoreboard was also displayed in the game view. Thus, using the JavaScript implementation the component was easy to transfer to the additional pages as well.

### Game view

The game view implements the game. Users were shown a video and a question with an annotation example on top of the video. Below the video current time of the video, the progress of the game, and hotkeys were displayed. Users could mute the sound, pause the video, and use ESC to cancel the drawing process. The game view is displayed at figure 3.3.

The video is displayed in HTML5 `<canvas>` element. This allows manipulation of the frame, which is used to allow users to draw on the top of the video. A video element is created dynamically, then event listener for "loadeddata" is attached. This listener is triggered when the video is ready to be played. Inside this listener, we call the `requestAnimationFrame` which calls the function to draw the image of the video. The draw function then recursively calls itself for every frame of the video.

On the canvas event listeners are attached for `mouseup` and `mousedown` events. When the events occur the callback functions are called, which starts the drawing process. The drawing process is finished if the drawing was started inside the canvas and finished inside the canvas. The result is then

posted on the server.

The server returns with the results, informing if the user was correct or incorrect. If the user was correct the server returns a list of correct and incorrect boxes used to calculate the results. The correct and incorrect boxes were displayed to the user to promote cooperation among the users.

Inside the main loop, which controls the user input and communication, three important controllers were implemented: *progress*, *sound*, and *statistics*. The game view modules and general overview can be seen in the figure 3.6. The controllers are small modules responsible for little tasks inside the game.

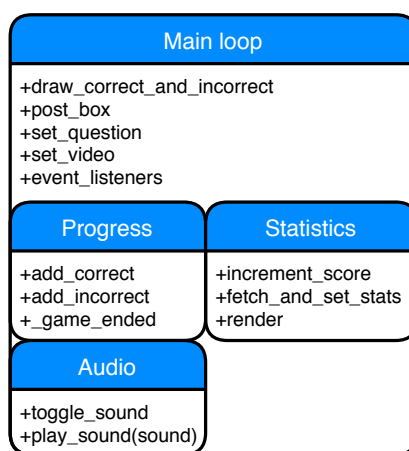


Figure 3.6: Game view modules

The audio controller is responsible for the sound elements of the game. The server responds to the client's answer with the correct sound identifier, indicating if the user was correct or incorrect. This information is passed to the audio controller. In addition to playing the sound, the controller calls the progress controllers *add\_correct* or *add\_incorrect* functions. In a sense, the audio controller is the first module in a pipe responsible for processing the results from the server.

Progress controller monitors the current games progress and displays it to the user. When the server returns with the correct or incorrect answer, the progress controller is informed. The controller then appends the progress bar of the game view as well as keeps count of the answers. The game is considered ended when the user has answered 20 questions. When the game is ended, the progress controller calls the render method of statistics controller.

The statistics controller is responsible for displaying the statistics at the end of the game. These statistics are displayed only for registered users.

Every time a user answers correctly the score variable is incremented inside the statistics controller. At the end of the game, the statistics controller posts the score to the server. The server responses with statistics about the user performance, which are then displayed to the user.

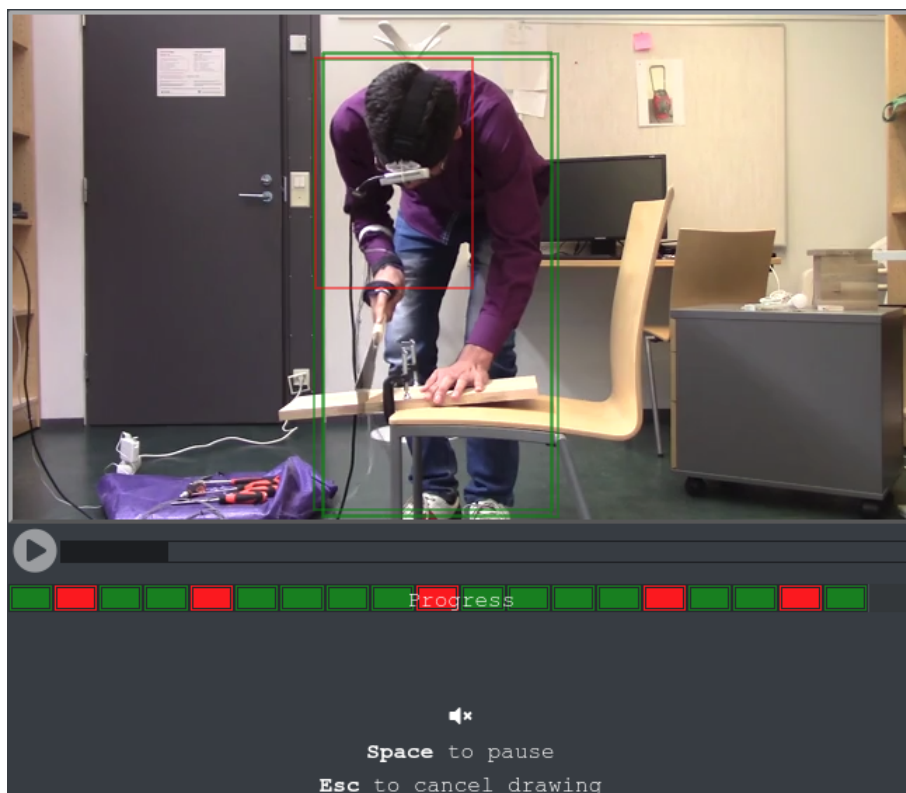


Figure 3.7: Previously given correct answers in green and incorrect answers in red are displayed to the user.

### Database management

There are two possible ways for modifying the existing database for administrator users. The first way is through Django's predefined view for database management. This view allows full access to the database.

A second view was generated for creating tasks and is displayed in figure 3.2. In this view, the video could be previewed at the same time, making it easier to determine a task object and duration. This gave the ability to quickly assess all the relevant tasks for time segments.

## Model preview

The model preview page enables the users to preview the votes accumulated on a specific video. Votes were displayed in a bar chart and updated over time. The view could be used to estimate the progress of the labeling process. A video could be removed when a desirable amount of answers were gathered. A bar chart of the video used in the user study is displayed in the figure 3.8.

The view also includes a video preview in which the user could display the annotations created from the crowdsourcing. The annotations were gathered from the server and linear interpolation was used to determine the box coordinates in a specific time.

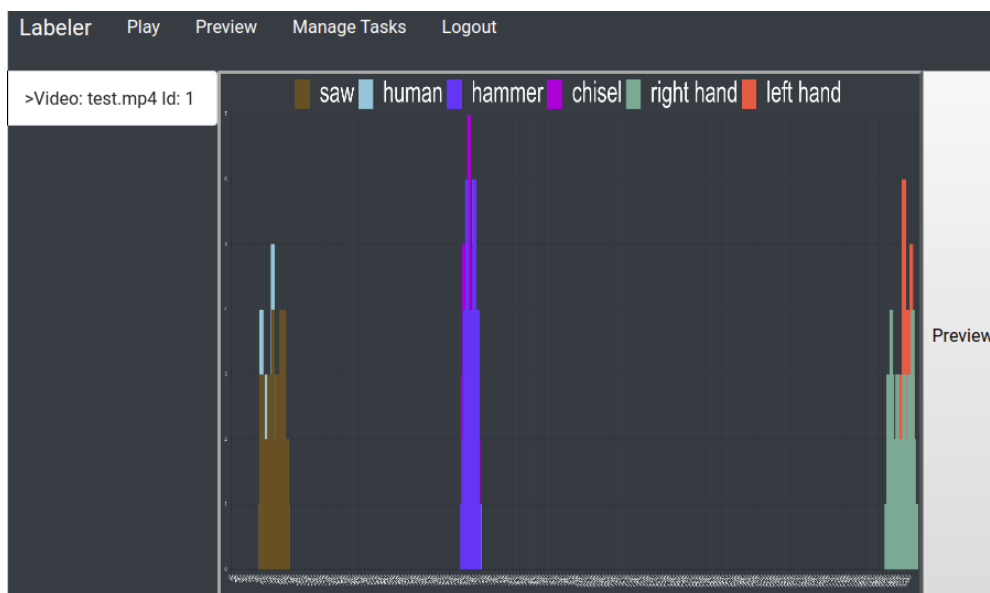


Figure 3.8: Model preview page displays the number of answers in points of time at the video. The video can be previewed with the annotations by clicking the preview button on the right.

## Server

The server's responsibility is to manipulate the data and provide views for the users. This architectural pattern is commonly described as model-view-controller (*MVC*). In addition to rendering the views and sending them directly to the users, the server also provides an application program interface (*API*) endpoint as an additional communication channel. The API is implemented following the representational state transfer (*REST*) principles.

The users that are flagged as superusers or admins are capable of creating and uploading videos, questions, and tasks. A task is a combination of a video segment with a question. These tasks are given to users. The users are then asked to provide answers to specific tasks. Based on these answers we built the annotation models.

The primary responsibilities of the server are database management, access control, and template rendering. Most of the communication was done through the REST API. The client uses the REST API to communicate with the server in the background without loading the page repeatedly. All the messages go through the NGINX reverse proxy.

## REST API

In REST architecture client can access and manipulate data with predefined stateless operations. The data is identified by uniform resource identifiers (*URI*) and the client can use GET, PUT, POST and DELETE HTTP methods to manipulate the data [42].

We wanted to create a interactive layout for the game so we decided that most of the communication should be done asynchronously. For this reason, a RESTful API was implemented. The three endpoints for the API were *highscores*, *tasks* and, *annotations*.

Task endpoint was the core of the whole game process. It has a methods for handling incoming POST requests as well as GET requests from a user. When the GET method was provoked the server fetches and creates a random task for the user. A video clip was produced with the combination of a specific task, as explained in more detail at chapter 3.2.2.3. For users flagged as admins, there is also an endpoint which can be used to create tasks for specific videos. This was used in the task edition view.

Annotations endpoint was used to fetch and convert the answers to JSON format. The endpoint also provided the statistics for the chart displayed to logged users. The chart could be used to count the answers in specific time points of a video. This way the users could get an overview of the annotation progress.

The high scores endpoint implemented the leaderboards and user statistics. After every successful game the client would send the results to the server and it would save them. Users could also request other users profile information: username, level, accuracy, contributions, and credits. This was done by clicking the user from the scoreboard.

## Database

Most of the data is stored in a SQLite database which is controlled by the Django back end. The structure of the database is displayed in the figure 3.9. The Django's user model was extended with additional properties defining user accuracy, experience, level, and credits. This allowed us to display statistics for registered users. A task defines the question and the video segment relevant to the question. An answer is related to a task, which specifies the label(question) for the answer.

Additionally, videos are saved straight to the filesystem and a reference path to the file is saved in the SQLite database. The writing of video file to the filesystem is performed automatically by the Django back end. The field is declared as a FileField and keyword parameter of the location in the filesystem is given to the field. The Django is responsible for writing the files to the defined path when admins upload videos to the server.

In the local development version, the back end is serving the videos to the user. In the production version, the reverse proxy is responsible for accessing the file system and serving the video files to the user.

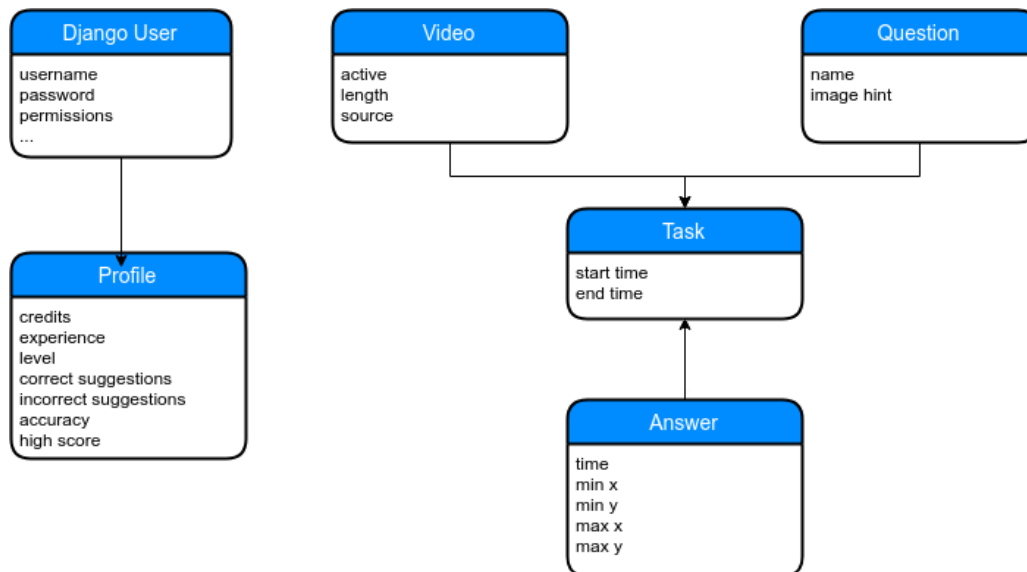


Figure 3.9: Database diagram

## Segmentation

The video used in the study was 33 minutes long. The server required the ability to cut the video to smaller segments. This segmentation was imple-

mented in two ways: one for local development and one for production. The local development version uses FFmpeg, a collection of libraries and tools to process multimedia content [22]. This approach was determined to be slow and inefficient. Every time a new clip was requested, the FFmpeg implementation created a new video from an existing one for the requested time frame. The production version uses the pseudo-streaming module that nginx provides. The pseudo streaming implementation was faster and handled the traffic easier.

Preprocessing of the video was required for it to be suitable for streaming. Video files do not store the complete image information in every frame. A frame can be defined by storing only the changes that occur from the previous frame. This encoding technique is used to compress video files. The frames that contain the whole picture are called keyframes. Keyframes are required in positions that are used to start the streaming, otherwise, the client could not render the image. For this reason, we had to add keyframes for every second of the videos. Additional keyframes increased the video size slightly.

## Chapter 4

# Evaluation

The game was evaluated from two points of view. Firstly, user satisfaction was measured using a questionnaire and considering the time spent on the page. Secondly, a ground truth model was created and the annotations from user answers were compared against the ground truth model.

### User study

Without a random unbiased sample from the target group there is no logical basis for generalizing the results to the target group. Thus, the important part of user study is to get a random sample of people in the group that the system is evaluated against. [49][5] First, we used common sense to create the target audience group. Underage users were excluded from the study due to ethical and lawful limitations. We characterized the common users, which we wanted to study, in the following way:

- **Age:** Users of legal age were allowed to participate, with the focus of users in ages 18 to 35.
- **Gender:** Puzzle type games are more popular among women [20]. Contrary to popular belief of male dominance in gaming, we expect an equal distribution of female and male users [57].
- **Environment:** Games are usually played in leisure time, thus the environment should be relaxed and enjoyable.
- **Skills:** Basic technological knowledge of computer use and varying experience with games.

## Pilot test

A pilot test was conducted before the actual user study, to verify that the task and the instructions were clear. The pilot study was conducted with three users. These participants were not used in the larger study, which was conducted after the pilot study. In the pilot study, users were asked to perform the task and give free feedback on any difficulties when performing the tasks. Open-ended questions were asked about the clearance of instructions and the usability of the system. The participants had close relationships with the moderator, which can introduce bias to the study [44]. However, we reasoned as the intention of the study was to determine any faults on the system, these biases would not affect the experiment.

## Users task

The task consists of two parts: the game playing and the questionnaire part. First, the users were asked to play a round of the game. Instructions for the game were displayed in the beginning. Secondly, if the user had not answered the questionnaire before, they were asked to fill out the survey.

We chose to use The System Usability Scale (*SUS*), as it is an industry standard method for measuring usability. The *SUS* has been quoted as "*quick-and-dirty*", containing only 10 questions, making it effortless to answer. [49] The *SUS* produces a single reference score that can be used to analyze the system, as such, the individual questions are not useful on their own [4]. The calculation is defined by Brooke [7]:

1. Sum all score contributions. For items 1,3,5,7 and 9, the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position.
2. Multiply the sum of the scores by 2.5 to obtain the overall value of *SU*.

In addition to the *SUS* questionnaire, we asked about the background of the users and their enjoyment of the game. Three statements were evaluated with a scale from one to five to evaluate user enjoyment. These statements were: *I liked the game*, *the game promotes cooperation and/or competition among the players* and *the game becomes monotonous as it progresses (repetitive or boring task)*. The whole questionnaire can be found at the end of this thesis.

## Subject group

On the user study, 20 subjects were gathered. Most of the subjects were between 18 and 35 of age, only one subject was older. The subjects were distributed fairly equally in gender, 65 percent of the subjects were male and 35 percent were female. In the figure 4.1 we can see the subjects experience with games. One-third of the users answered the questionnaire at home and one-third of the users answered in their office. The remaining participants did not answer the question about the location.

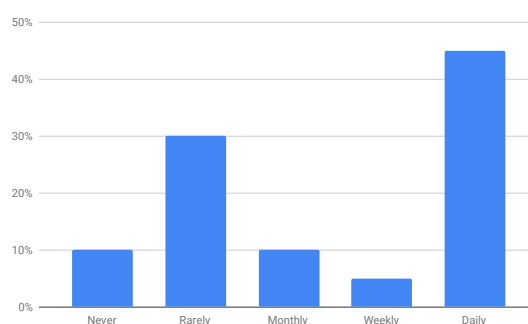


Figure 4.1: How often the subjects played games.

The subjects were representing the target group fairly well, there were little less female subjects than expected. Also, the fact that half of the users did the questionnaire at the office was surprising. The effect of recruiting method had an impact on this, one of the subjects recruited his co-workers at their office to do the study.

Some of the individuals in the user study were associated with the moderator. As discussed this can introduce positive bias to the study and the bias was noticed during the study. The bias is studied more in detail in the following chapters. The rest of the subjects were recruited using chain sampling, the recruited personnel were asked to recruit more people to the study.

## Results

The average SUS value for the user study was *70,5*. It is important to note that the SUS score is not to be used as a percentile and the individual questions are not meaningful on their own. Studies analyzing the SUS score has shown that the score *70,5* ranks on the high acceptance ratings. This score can be described as the adjective good, within the margin of errors.[3, 4]

When subjects were asked if they liked the game, the answers were heavily weighted towards the satisfied rankings. However, on the first inspection monotonous or boring games are rarely thought out as a good game. As we can see from the figure 4.2 the results are highly contradictory with the question *"The game becomes monotonous as it progresses (repetitive or boring tasks)"*. The result was somewhat expected as the question of liking the game was on many occasions brought to a topic after the tests. Few individuals even mentioned that they gave a lenient ranking on the question: *"I liked the game"*.

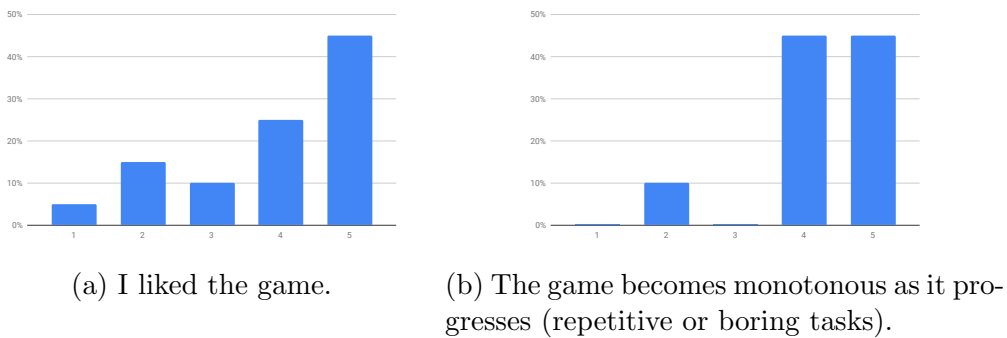


Figure 4.2: Contradictory answers for questions 11 and 13.

An effective method for evaluating user engagement is voluntary participation. We compared the number of games played to the number of participants in the study. We also analyzed the distribution between unregistered and registered users. In total the game was played 22 times across 20 subjects. This shows that most of the users did not get really engaged in the task as they were gone after the initial try. In addition, six users created their own profiles. Two reasons for creating accounts were noticed. One group of subjects wanted to be shown on the ranking boards. The second group wanted to inspect the existing models created from the game.

One goal of the game was to promote cooperation among the players to create a feeling of satisfaction. In the questionnaire, we asked the subjects to evaluate their opinion on the promotion and the answers can be seen in the figure 4.3. The answers were fairly scattered, this can be a result of multiple different variables. In some extent the experience of the game mattered, some of the individuals were more intrigued by the game and asked more questions before the study.

These individuals usually learned that the game was created in the purpose of crowdsourcing the task, thus, cooperation was more visible for them. Furthermore, the user study was conducted in a fairly small manner which

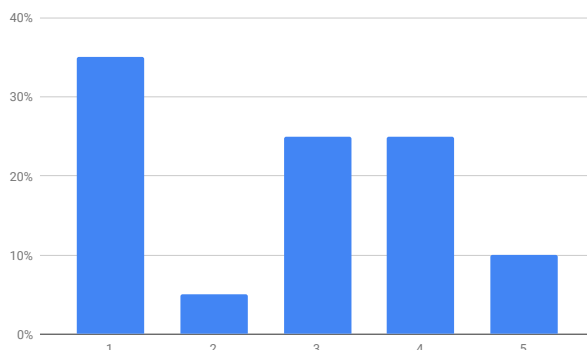


Figure 4.3: The game promotes cooperation and/or competition among the players.

makes the margin of error greater.

## Label accuracy

After the study was conducted all the answers from the game were gathered. These answers were used to build a bounding box model for every task. A ground-truth model was built by playing the game. We played the game until the preview of the models were empirically found to be sufficiently accurate. The ground-truth model was used to analyze the accuracy of the model built by the users in the study. For the accuracy measurement, we used intersection over the union, described in the chapter 3.1.2.1.

We used seven different tasks in the study. These tasks had different difficulties. The objects varied in size and movement. For some of the tasks, like following a hammer while hammering, the model building process was hard as the frame rate of the video could not keep up with the hammer. The tasks were in following three different time segments:

1. First segment of 87 seconds. Users were asked to track a saw and a human.
2. Second segment of 54 seconds. Users were asked to track for a chisel, hammer and the right hand of the human.
3. Third segment of 90 seconds. Users were asked to track for the left and right hands of a human.



Figure 4.4: Example frame from the second segment.

## Interpolation

Interpolation is a method for estimating new data points within known data points. Linear interpolation was used to calculate the bounding boxes for frames that were not annotated by any user. The bounding boxes are defined by  $x_1, y_1, x_2, y_2, time$ . The coordinates define the largest and smallest values for coordinates  $x$  and  $y$ , i.e., the upper left corner and the bottom right corner. The interpolation was performed separately for every point in the bounding box. To interpolate point  $x$  in time  $t$  the following equation 4.1 was used:

$$x = x_a + (x_b - x_a) \frac{t - t_a}{t_b - t_a} \quad (4.1)$$

## Post-processing filters

The user provided answers included a lot of noise. To combat this problem two filters were implemented. Both of the filters use the area of the bounding box. The area was calculated using the formula 4.2. In the first filter, only the area was considered. The first filter removes all zero area bounding boxes. As a result, the miss clicks from the users were filtered.

The second filter used mean and standard deviation to perform the filtering. First the mean and standard deviation of the area were calculated using the standard formulas:

$$A = (x_2 - x_1) * (y_2 - y_1) \quad (4.2)$$

$$\bar{A} = \frac{1}{n} \sum_1^n A_n = \frac{A_1 + A_2 + \dots + A_n}{n} \quad (4.3)$$

$$\sigma = \sqrt{\frac{\sum_1^n (A_i - \bar{A})^2}{n - 1}} \quad (4.4)$$

Then all answers in a task were iterated and compared using the formula 4.2.2. All the items satisfying the constraint were kept. The default value used for  $\alpha$  was one, but different values for  $\alpha$  were tested.

$$|\bar{A} - A| < \alpha \sigma \quad \alpha \in R \quad (4.5)$$

## First segment

The highest accuracy was achieved in the first segment, as it was the most simple one. Both of the tracked objects were rather still. The human was large and the saw was small. However, even though the first segment was quite easy, there was a lot of noise in the answers. Figure 4.5 displays the IoU over time for the unfiltered annotations in the first segment. The unfiltered data shows that there is a lot of noise in the models built by the users. The spikes to zero accuracies were mostly resulting from miss clicks from the users. Applying the area filter improved the accuracy vastly. The area filter removed all the bounding boxes which had the property of  $x1 = x2$  or  $y1 = y2$ .

By inspecting the video, we realized that there were a lot of poor annotations which were too small or too large. Thus, we applied the standard deviation based filtering after the miss clicks were removed. The standard deviation filtering helped to clear the data and for human tracking, the results were good and suitable for machine learning purposes. However, for the saw the resulting accuracy was not desirable. By inspecting the bounding boxes in the saw model, we noticed that users opted for drawing overall larger boxes. This resulted in lower accuracy as the boxes were mostly larger than in our ground-truth model. However, the location of the saw could be determined quite accurately, as the center points of the boxes are close in distance in both models.

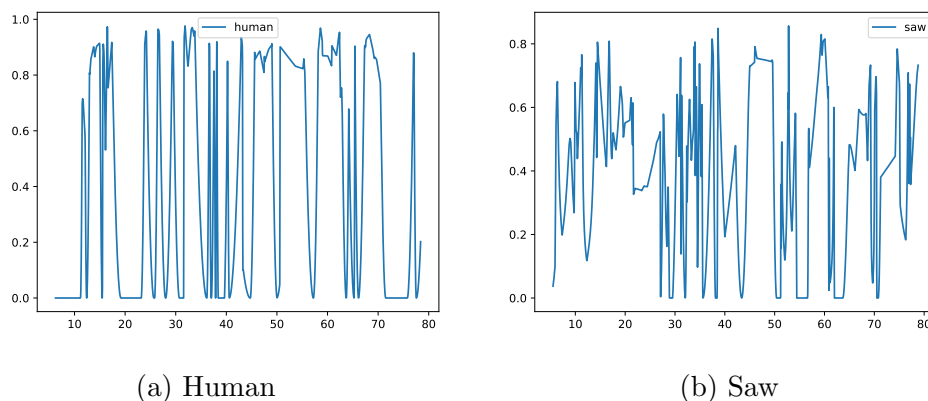


Figure 4.5: Unfiltered tracking of two objects in the first segment. Intersection over the union in the y-axis and time in the x-axis.

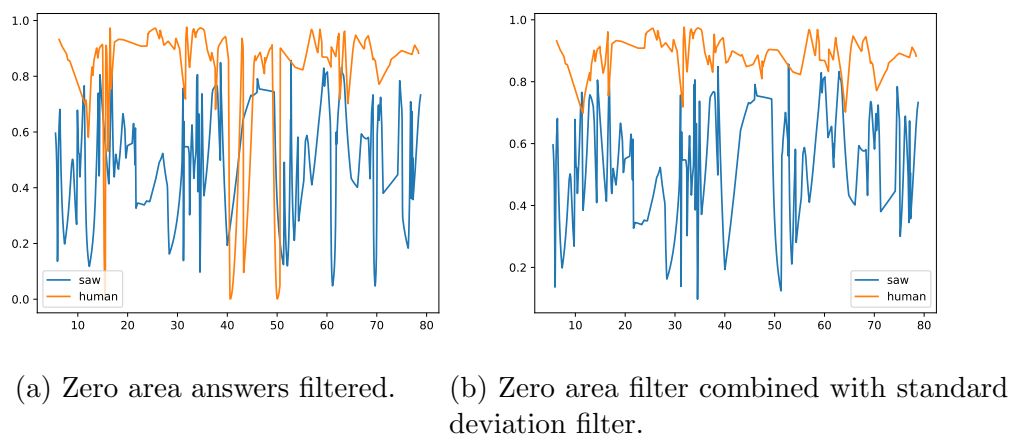
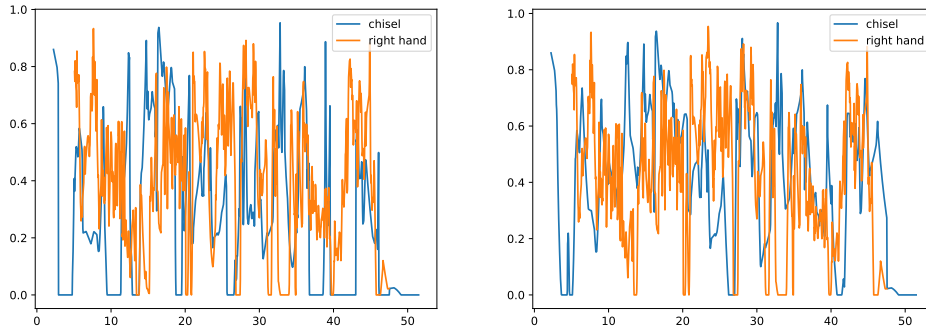


Figure 4.6: Two different filter techniques applied on the first segment.



(a) Zero area answers filtered. (b) Zero area answers filtered combined with standard deviation filter.

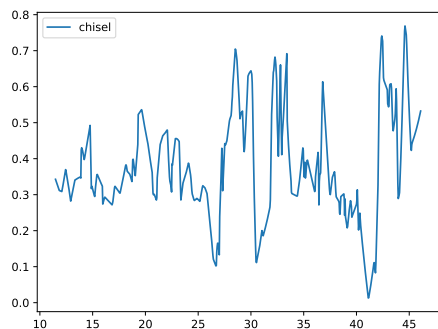
Figure 4.7: Filtered tracking of two objects in the second segment.

## Second and third segment

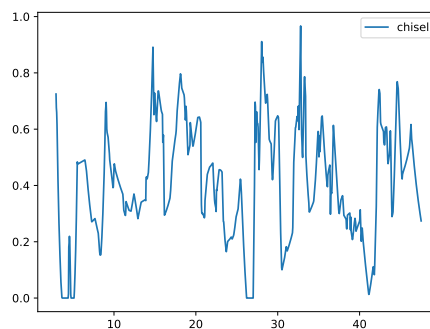
The objects for the second and the third segment were much more difficult than the ones in the first segment. The objects moved faster and changed size as well. The ground truth model for the second segment was built for the right hand and the chisel. To build an accurate ground truth model over 200 answers were inserted to the game. In contrast, the first segment only needed 23 answers for the saw and 35 answers for the human to achieve a good ground truth model. From the user study, 80 answers were gathered for the right-hand model and 89 answers were gathered for the chisel model.

Different values for the  $\alpha$ , defined in equation 4.2.2, were experimented with in the second segment. For the right hand, the results seemed to be unaffected by the different alpha values. The chisel model increased in accuracy with lower alpha values. After filtering using  $\alpha = 0.5$  20 answers remained. Respectively using the  $\alpha = 0.8$  41 answers remained. The chisel object was stationary in many of the frames, thus by the remaining boxes with similar size achieved the best results.

In the third segment, we built a ground truth model for the right hand. We inserted 200 answers to build the ground truth, which was twice the amount users gave us. As expected, the unfiltered data was noisy. We implemented an algorithm which searched the best alpha value with 0.01 precision. With value  $\alpha = 0.69$  highest accuracy was achieved. The figure 4.9 displays the accuracy for the optimal  $\alpha$  value in the last segment. Thus, even with the optimal alpha value the score was undesirable.

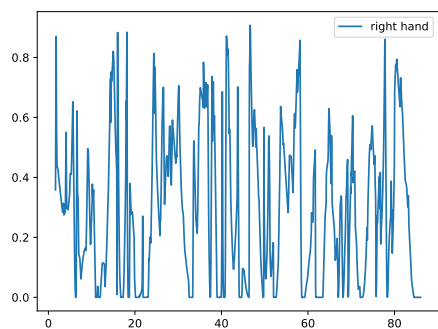


(a)  $\alpha = 0.5$ .

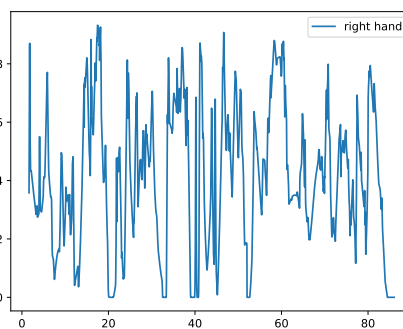


(b)  $\alpha = 0.8$ .

Figure 4.8: Different  $\alpha$  values used in standard deviation filter to the chisel model.



(a) Unfiltered



(b) Best value pruned for alpha.  $\alpha = 0.69$

Figure 4.9: Right hand of last segment.

## Summaries

Almost all the users could play the game without additional information. This result also was backed by the fact that the SUS score ranked well, the system was thought to be easy to use.

Overall the game did not fulfill the requirements. The engagement for the game seemed to be nonexistent. Even though the game achieved high rankings in the general question about the users liking of the game, most of the users were not staying on the page after they answered the questionnaire. The game was also ranked highly repetitive and boring. As follows, the results for the game enjoyment were considered mostly from the bias of the user study group. Some of the users were distinctly related to the developer.

The data from the users was not suitable for building the model. The data included a lot of noise and the users generally draw inaccurate boxes around the objects. Applying the two filters helped to clean the data. Only the human in the first segment resulted in good accuracy, staying close to 80% accuracy in the whole time. As a reference, in the PASCAL VOC challenge predictions resulting in 50% accuracy were considered detected [21].

One problem with the survey is that the number of answers gathered was limited. During the construction of the ground truth models we inserted more answers to the complex segments, like the second segment, than we got from the user study. Taking in to account the amount of bad answers users gave us, it is possible that the accuracy could be improved in the models by gathering more answers from the users.

Even though we could search for the most optimal value for the standard deviation filter, the results were not desirable. However, the filters seemed to be efficient. In all cases, they improved accuracy. This could change if the tracked object changed size during the video.

Even though the accuracies were not desirable the data could still be useful in the sense of machine learning. Building a straight forward object detection model could be difficult, as the bounding boxes were not that accurate. However, for different purposes information about the general location and movement of the objects could be gathered.

## Chapter 5

# Discussion

Our user study showed that the game did not fulfill the requirements and users were not interested in the game. We encountered a lot of different small problems and this chapter discusses the problems and possible solutions. In addition, insight from the study is displayed and suggestions for future work are given.

### Possible evaluation distortions

It is hard to exactly estimate the target audience of the developed game. As the game was fairly simple and monotonous, the target audience could be different from our estimation, for example, smaller children. From our user study, we can not generalize results for this group, as they were excluded due to ethical and lawful limitations.

The ground truth model was built using the local development version, which uses the FFmpeg to cut the videos. The deployed web page uses nginx pseudo-streaming module to cut the videos. It was noticed that in the last segment it seemed like the ground truth model was few milliseconds off from the actual video. This could affect the results of the study. The distortion was not thoroughly examined, as by empirical analysis the results were estimated to be adequately accurate.

The user study group was fairly small, only 20 users were recruited. The group had also people who were acquainted with the study conductor. This can introduce bias to the study as previously discussed, and was also noticed in the questionnaire answers.

## Moving towards crowdsourcing platforms

In the user study, some attraction from the game elements was noticed. Some of the subjects created accounts in the sole purpose of being displayed in the leader boards. This kind of extra attractions through gamification could be helpful on the crowdsourcing platforms. By giving the workers incentive outside the monetary realm, the task providers could try to lower the costs. The additional incentive could help to provide more accurate labels, as our study shows, the bounding boxes were rather inaccurate and similar results are reported in the crowdsourcing platforms [56].

Our game was not studied in the crowdsourcing platforms. It is important to note that the game should be as efficient as possible if the considered use is inside a crowdsourcing platform. Our game used random sampling and users were free to give the answer at any point of time, resulting in grouped answers in easier frames. This was not considered a problem in the implementation if the users would consider playing the game freely. In contrast, this kind of inefficiency would raise costs when users are paid to answer.

To achieve this efficiency the system would need to be designed in a different way. One possible solution would be to display the user two different boxes from previous users and asked to draw a box to a frame between the two answers. This could also make the game more fun and make the users more curious about the game. This would also prevent users from marking the same frames. Also, the users could be restricted to be working only on images. Making it easy for the server to determine which frames are marked and which ones are not.

## Improving the used methods

The game would benefit from a more dynamic approach for the task selection as well as the determination of questions. To make the game more efficient at producing the labels the system would need to track and provide tasks in a manner that distributes the answers in the best way. The algorithm should take to an account the current density of answers and the difficulty of the segment, fast-moving objects need more answers to provide accurate results.

Determining if the user was correct or incorrect was done by searching for a fixed sized window in time-space and comparing these answers to the user-provided answer using intersection over the union. At the end of the user study, many of the answers were determined to be incorrect when playing the game. This was probably due to the window size being too large.

The server should be able to estimate the difficulty when choosing the

time window. It would be possible to estimate the movements of an object by inspecting previous answers distances. If the answers would be scattered, the server should opt to use a smaller window of time.

The filters used in the user study provided good results. The drawback of these filters is that they only consider the area over the whole segment. For objects that can change in size as a result of moving further away or rotation, the algorithms are suspected to work poorly. In our user study the average area of every object was quite constant, so they worked well in the study.

Working with the video elements in HTML could not provide the same frame level accuracy as the image extraction would. This was also noticed in the ground truth model building process, overall the models were slightly off in the videos. Some of the tools for bounding box annotation combat this problem by breaking down the video to images [56][50]. A pseudo video player can be implemented by displaying a stream of images to the user. The downside of this approach is that images require more space and processing from the back-end.

## Game design

A few years back the game industry was considered a niche market. Now the game industry is a multi-billion dollar industry. Average American citizen spends more time playing games than going to the movies. Due to the new highly competitive nature of the gaming industry, games need to be well designed and marketed. [38]

Some of the previous successful labeling games like Google's image labeler were launched in early 2000. Many of the new labeling methods focus on the crowdsourcing aspect. This is most likely due to the increased complexity and competition in the game market. It is not easy to build a game that satisfies the users, as there is an abundance of games in the market today. Even the arguably largest influencer in the Internet, Google, halted the labeler game for a brief period.

Designing a game and implementing it in a appropriate time limit was a difficult task and a lot of improvements can be done in the game design aspect. Some of the requirements for the game were discussed in the chapter 3.1.1. The following paragraphs are dedicated to different approaches for the game design.

Many of the phone games are simpler in nature, making them attractive for crowdsourcing small tasks. People also tend to play games for a short period of time, for example, while waiting for a bus. Thus, moving the

game platform towards mobile could relax the constraint manifested by the repetitive nature of the labeling process. Currently, mobile games are the most fast growing gaming section [38].

For many annotations, it would be possible to work only with images. Working with images is lightweight compared to videos. This could open up new possibilities for building a game. Users could be introduced dragging abilities and multiple images could be displayed to users. Puzzle games similar to popular game Bejeweled blitz could be implemented, in which users are asked to align same type of pictures together. This could allow the grouping of different images in the same categories.

Implementing a memory based game, in which users are made to watch a segment of a video and answer questions based on the videos could make the gameplay more interesting. With appropriate design, the answers could be used to label the data.

For videos, a dragging game could be implemented, in which the users were asked to hold down the cursor and follow a specific item. Constantly rewarded points for correctly following the object. These kinds of outputs could be processed in a way that they help to build models around holistic representations, explained in the chapter 1.1.

One major challenge of the game design was the fact that some kind of information is needed to determine if the user's answers were correct or incorrect. In our implementation, we used an approach of comparing the previous answers to the currently provided answers. However, this approach does not work when there are no answers available for the current task. For our game, this does not provide a real difficulty, as the answer was then automatically rewarded as a correct one. For a puzzle like games this could be a major drawback, if the same type items need to vanish, it is sometimes impossible to determine if the user made a correct or incorrect move. This could hinder user satisfaction greatly. To combat this some kind of machine learning could be implemented, for example, in the dragging game optical flow of the video could be used to determine if the user is moving to the correct way.

## Chapter 6

# Conclusions

Labeling videos with accurate bounding box labels take a lot of man power. However, this kind of annotations are useful in machine learning. Many large datasets exist, but they are usually labeled coarsely. In other cases there is no existing database to be used and manual work is required.

Crowdsourcing has played a big role in data mining. Labeling for computer vision is an especially attractive task for crowdsourcing as the task is simple to break down to smaller parts. These parts can be worked in parallel and the results can be aggregated easily. In recent years many specialized crowdsourcing platforms have risen. These crowdsourcing platforms offer monetary incentive to recruit workers for tasks.

The core problem of the thesis is the masking of a monotonous and repetitive task in a clever way. The question is straightforward: *How can we get the users to enjoy this monotonous and repetitive task?* Multiple different game concepts were thought out and in the end we implemented a simple game in which users were asked to draw boxes around objects in videos. The users were also able to create accounts to be shown in leader boards and track their progress with levels and experience.

After the implementation we conducted a user study to evaluate the game and usability of the system. To analyze the annotations from the user study we created a ground truth model. Using intersection over union as accuracy measurement we calculated the accuracy of the annotations. With two different post processing filters higher accuracies were achieved.

Most of the users did not continue playing the game after the initial try. There were 20 subjects in the user study and 22 rounds of the game were played. The questionnaire reveals that the game was thought to be repetitive and boring. Using the system usability scale we learned that the game was easy to use and users could play the game with the instructions displayed on the page.

Few of the subjects registered to the website. Some of them out of curiosity and others for the purpose of being displayed in the leader boards. This clearly shows that the game elements can increase the engagement for a task in hand. However, studies show that users with an incentive to work the task freely benefit the most for the game elements [26]. As stated, most of the users were not really interested to freely spend time on the game.

Analyzing the bounding boxes from the game we realized that the dataset consists of a lot of noise and inaccurate answers. Two post processing filters were applied to filter out the noise. We used one filter to remove all bounding boxes with zero areas. The second filter calculated the mean of area  $\bar{x}$  and standard deviation  $\sigma$ . Then the distance between the mean in correlation to the standard deviation was used to determine if an answer was kept or filtered out. Combination of these two filters was found to create the most accurate results.

Good results were achieved in one model, which consists of a large bounding box over a mostly stationary human. Other models varied in accuracy, most of them were not sufficiently accurate. As a reference point we used the PASCAL voc measurement threshold, intersection over the union of 50% for sufficient accuracy [21].

Overall the study shows that little attraction was achieved using the game elements on the annotation process. In addition, there are a lot of possible improvements to the game. These were not carried out as the game design was thought to be flawed. We tried to find an answer for masking the annotation process. This requirement was not fulfilled.

For future work integrating game elements to the crowdsourcing platforms could be beneficial. With an efficient game to build the annotations, the game creators could try to offer lower monetary reward than the other competitors. This could lower the costs of the labeling process. It is important to note that the game would need to be constructed in an efficient manner, in a way that as little as possible answers are needed to build the annotations. In the current state our game is too inefficient to be deployed in these crowdsourcing platforms.

# Bibliography

- [1] *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012* (2012), IEEE Computer Society.
- [2] ABU-EL-HAIJA, S., KOTHARI, N., LEE, J., NATSEV, P., TODERICI, G., VARADARAJAN, B., AND VIJAYANARASIMHAN, S. Youtube-8m: A large-scale video classification benchmark. *CoRR abs/1609.08675* (2016).
- [3] BANGOR, A., KORTUM, P., AND MILLER, J. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Studies* 4, 3 (May 2009), 114–123.
- [4] BANGOR, A., KORTUM, P. T., AND MILLER, J. T. An empirical evaluation of the system usability scale. *Int. J. Hum. Comput. Interaction* 24, 6 (2008), 574–594.
- [5] BEVAN, N., AND USABILITY SERVICES, S. Common industry format usability tests.
- [6] BORIS SEKACHEV AND NIKITA MANOVICH AND ANDREY ZHAVORONKOV. Computer vision annotation tool, 2018. WWW <https://github.com/opencv/cvat>. Accessed 19 October 2018.
- [7] BROOKE, J. Sus: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).
- [8] CHARLAND, A., AND LEROUX, B. Mobile application development: web vs. native. *Commun. ACM* 54, 5 (2011), 49–53.
- [9] CHATZIMILIOUDIS, G., KONSTANTINIDIS, A., LAOUDIAS, C., AND ZEINALIPOUR-YAZTI, D. Crowdsourcing with smartphones. *IEEE Internet Computing* 16, 5 (2012), 36–44.

- [10] COOPER, S., KHATIB, F., TREUILLE, A., BARBERO, J., LEE, J., BEENEN, M., LEAVER-FAY, A., BAKER, D., POPOVIC, Z., AND PLAYERS, F. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756–760.
- [11] CORDTS, M., OMRAN, M., RAMOS, S., REHFELD, T., ENZWEILER, M., BENENSON, R., FRANKE, U., ROTH, S., AND SCHIELE, B. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (2016), IEEE Computer Society, pp. 3213–3223.
- [12] CSE GROUP, MICROSOFT. Visual object tagging tool, 2019. WWW <https://github.com/opencv/cvat>. Accessed 4 April 2019.
- [13] DAMEN, D., DOUGHTY, H., FARINELLA, G. M., FIDLER, S., FURNARI, A., KAZAKOS, E., MOLTISANTI, D., MUNRO, J., PERRETT, T., PRICE, W., AND WRAY, M. Scaling egocentric vision: The EPIC-KITCHENS dataset. *CoRR abs/1804.02748* (2018).
- [14] DASIOPOULOU, S., GIANNAKIDOU, E., LITOS, G. C., MALASIOTI, P., AND KOMPATSIARIS, Y. A survey of semantic image and video annotation tools. In *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution - Bridging the Semantic Gap*. 2011, pp. 196–239.
- [15] DENG, J., DONG, W., SOCHER, R., LI, L., LI, K., AND LI, F. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA* (2009), IEEE Computer Society, pp. 248–255.
- [16] DIFALLAH, D. E., CATASTA, M., DEMARTINI, G., IPEIROTIS, P. G., AND CUDRÉ-MAUROUX, P. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015* (2015), A. Gangemi, S. Leonardi, and A. Panconesi, Eds., ACM, pp. 238–247.
- [17] DOCKER, INC. Docker documentation, 2003. WWW <https://docs.docker.com/>. Accessed 22 October 2018.
- [18] EDWARDS, S. M., FLANNIGAN, W. C., AND EVANS, P. T. 6-dof pose estimation: the need for standardization in industrial applications.

- In *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop, PerMIS 2010, Baltimore, Maryland, USA, September 28-30, 2010* (2010), E. Messina and R. Madhavan, Eds., ACM, pp. 267–270.
- [19] EICKHOFF, C., AND DE VRIES, A. How crowdsourcable is your task? *Mathematical Structures in Computer Science - MSCS* (01 2011).
- [20] ENTERTAINMENT SOFTWARE ASSOCIATION. Essential Facts about the computer and video game industry, 2018. WWW [http://www.theesa.com/wp-content/uploads/2018/05/EF2018\\_FINAL.pdf](http://www.theesa.com/wp-content/uploads/2018/05/EF2018_FINAL.pdf). Accessed 2 February 2019.
- [21] EVERINGHAM, M., ESLAMI, S. M. A., GOOL, L. J. V., WILLIAMS, C. K. I., WINN, J. M., AND ZISSERMAN, A. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision* 111, 1 (2015), 98–136.
- [22] FFMPEG DEVELOPERS. FFmpeg tool, 2018. WWW <https://ffmpeg.org/>. Accessed 5 March 2019.
- [23] GEIGER, A., LENZ, P., AND URTASUN, R. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012* [1], pp. 3354–3361.
- [24] GRIFFIN, G., HOLUB, A., AND PERONA, P. Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology, 2007.
- [25] GUO, X., WANG, H., SONG, Y., AND HONG, G. Brief survey of crowdsourcing for data mining. *Expert Syst. Appl.* 41, 17 (2014), 7987–7994.
- [26] HAKULINEN, L. Gameful approaches for computer science education: From gamification to alternate reality games, 2015.
- [27] HEILBRON, F. C., ESCORCIA, V., GHANEM, B., AND NIEBLES, J. C. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015* (2015), IEEE Computer Society, pp. 961–970.
- [28] HERATH, S., HARANDI, M. T., AND PORIKLI, F. Going deeper into action recognition: A survey. *Image Vision Comput.* 60 (2017), 4–21.

- [29] HUNICKE, R., LEBLANC, M., AND ZUBEK, R. Mda: A formal approach to game design and game research. In *In Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence* (2004), Press, pp. 1–5.
- [30] JUNG, A. A gentle introduction to supervised machine learning. *CoRR abs/1805.05052* (2018).
- [31] KIM, E., HELAL, S., AND COOK, D. J. Human activity recognition and pattern discovery. *IEEE Pervasive Computing* 9, 1 (2010), 48–53.
- [32] KOIVISTO, J., AND HAMARI, J. The rise of motivational information systems: A review of gamification research. *International Journal of Information Management* 45 (2019), 191 – 210.
- [33] KONG, Y., AND FU, Y. Human action recognition and prediction: A survey. *CoRR abs/1806.11230* (2018).
- [34] KWAK, D., KAM, A., BECERRA, D., ZHOU, Q., HOPS, A., ZAROUR, E., KAM, A., SARMENTA, L., BLANCHETTE, M., AND WALDISPÜHL, J. Open-phylo: a customizable crowd-computing platform for multiple sequence alignment. *Genome Biology* 14, 10 (Dec 2013), R116.
- [35] KWON, S., AND CHA, S. D. Captcha-based image annotation. *Inf. Process. Lett.* 128 (2017), 27–31.
- [36] LECUN, Y., AND CORTES, C. MNIST handwritten digit database.
- [37] LEE, J., KLADWANG, W., LEE, M., CANTU, D., AZIZYAN, M., KIM, H., LIMPAECHER, A., YOON, S., TREUILLE, A., DAS, R., AND . Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences* (2014).
- [38] MARCHAND, A., AND HENNIG-THURAU, T. Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities. *Journal of Interactive Marketing* 27, 3 (2013), 141 – 157.
- [39] MATZEN, K., AND SNAVELY, N. Nyc3dcars: A dataset of 3d vehicles in geographic context. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013* (2013), IEEE Computer Society, pp. 761–768.
- [40] MD OSMAN, G. A novel approach to complex human activity recognition, 2017.

- [41] MORSCHHEUSER, B., AND HAMARI, J. The gamification of work: Lessons from crowdsourcing. *Journal of Management Inquiry* 0, 0 (0), 1056492618790921.
- [42] MUMBAIKAR, S., PADIYA, P., ET AL. Web services based on soap and rest principles. *International Journal of Scientific and Research Publications* 3, 5 (2013), 1–4.
- [43] PRESTI, L. L., AND CASCIA, M. L. 3d skeleton-based human action classification: A survey. *Pattern Recognition* 53 (2016), 130–147.
- [44] RIIHIAHO, S. Experiences with usability testing: Effects of thinking aloud and moderator presence, 2015.
- [45] ROHRBACH, M., AMIN, S., ANDRILUKA, M., AND SCHIELE, B. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012* [1], pp. 1194–1201.
- [46] RUBINO, C., FUSIELLO, A., AND DEL BUE, A. Lifting 2d object detections to 3d: A geometric approach in multiple views. In *Image Analysis and Processing - ICIAP 2017 - 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part I* (2017), S. Battiato, G. Gallo, R. Schettini, and F. Stanco, Eds., vol. 10484 of *Lecture Notes in Computer Science*, Springer, pp. 561–572.
- [47] RYAN, R. M., AND DECI, E. L. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology* 25, 1 (2000), 54–67.
- [48] RYOO, M. S., AND AGGARWAL, J. K. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *IEEE International Conference on Computer Vision (ICCV)* (2009).
- [49] SAMPAIO, A. Quantifying the user experience: practical statistics for user research by jeff sauro and james r. lewis. *ACM SIGSOFT Software Engineering Notes* 38, 1 (2013), 57–58.
- [50] SHEN, A. Beaverdam: Video annotation tool for computer vision training labels. Master’s thesis, EECS Department, University of California, Berkeley, Dec 2016.

- [51] TASKULA, T. Advanced data fetching with graphql: Case bakery service; edistyksellinen tiedonhaku graphql:n avulla: tapaustutkimus leipuripalvelu. G2 pro gradu, 2019-03-11.
- [52] VESA, M., HAMARI, J., HARVIAINEN, J. T., AND WARMELINK, H. Computer games and organization studies. *Organization Studies* 38, 2 (2017), 273–284.
- [53] VON AHN, L., AND DABBISH, L. Labeling images with a computer game. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004* (2004), E. Dykstra-Erickson and M. Tscheligi, Eds., ACM, pp. 319–326.
- [54] VON AHN, L., AND DABBISH, L. Designing games with a purpose. *Commun. ACM* 51, 8 (2008), 58–67.
- [55] VON AHN, L., LIU, R., AND BLUM, M. Peekaboom: a game for locating objects in images. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006* (2006), R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, Eds., ACM, pp. 55–64.
- [56] VONDRICK, C., PATTERSON, D. J., AND RAMANAN, D. Efficiently scaling up crowdsourced video annotation - A set of best practices for high quality, economical video labeling. *International Journal of Computer Vision* 101, 1 (2013), 184–204.
- [57] WILLIAMS, D., CONSALVO, M., CAPLAN, S., AND YEE, N. Looking for gender: Gender roles and behaviors among online gamers. *Journal of Communication* 59 (12 2009), 700 – 725.
- [58] XIANG, Y., MOTTAGHI, R., AND SAVARESE, S. Beyond PASCAL: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014* (2014), IEEE Computer Society, pp. 75–82.
- [59] XIAO, J., EHINGER, K. A., HAYS, J., TORRALBA, A., AND OLIVA, A. SUN database: Exploring a large collection of scene categories. *International Journal of Computer Vision* 119, 1 (2016), 3–22.
- [60] YU, F., XIAN, W., CHEN, Y., LIU, F., LIAO, M., MADHAVAN, V., AND DARRELL, T. BDD100K: A diverse driving video database with scalable annotation tooling. *CoRR abs/1805.04687* (2018).

- [61] ZICHERMANN, G., AND CUNNINGHAM, C. *Gamification by Design - Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly, 2011.

# Appendix A

## Questionnaire

1= Strongly disagree, 2=Disagree, 3=Neither disagree nor agree, 4=Agree, 5=Strongly agree

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.
11. I liked the game.
12. The game promotes cooperation and/or competition among the players.
13. The game becomes monotonous as it progresses (repetitive or boring tasks).

14. Further comments? (Strengths, weaknesses)

**Background**

15. Gender?

- (a) Male
- (b) Female

16. Age?

- (a) 18-25
- (b) 26-35
- (c) 36 or older

17. How often do you play games?

- (a) Never
- (b) Rarely
- (c) Monthly
- (d) Weekly
- (e) Daily

18. Where did you try the game? (Home, Office..)