Automatic semantic analysis of gameplay videos of 'This War of Mine'

S.A.A. den Broeder

Supervisor: R.C. Veltkamp
Thesis number: ICA-4100905

Utrecht University, The Netherlands

February 22, 2019

Abstract

In recent years there has been an increase in published research on video games. However, few articles discuss semantic analysis of gameplay videos, which are available online on platforms such as YouTube or Twitch. A case-study was performed where a three-step approach is proposed to analyze a specific scene in videos of the game 'This War of Mine'. The first step is to detect the location at which the scene of interest takes place using a SVM with bag-of-visual-words histograms of SIFT features. Then convolutional neural networks are used to detect which scene takes place at that location and what choice the player makes during that scene.

Contents

1 Introduction

In 2017, the video game industry was estimated to be worth approximately 108 billion dollars [38, 8, 57]. The massive growth of the gaming industry is coupled with an increase in research on games and gameplay over the last 20 years or so. Game research has been approached from a multitude of perspectives. Radde-Antweiler et al. [47] give an overview of game research, primarily focused on game environments and religion in games. They also discuss Let's Play videos, which will be discussed further on in this section. Canossa [11] takes an approach using telemetry and describes methods to turn the outcome of the telemetry system into metrics, features and models. Jørgensen [26] investigated how audio supports the visual gameplay of an online multiplayer game. Lankoski and Björk [31] described how the formal analysis method, used in different fields such as archeology, literature and film analysis, can be applied to the domain of video games. The types of game research mentioned so far are primarily focused on commercial or entertainment games. A different type of games are serious games, which Mitgutsch and Alvarado defined as games that "intend to fulfill a purpose beyond the self-contained aim of the game itself" [41]. They developed the Serious Game Design Assessment Framework, which analyses serious games based on their purpose, content and information, mechanics, fiction and narrative, aesthetics and graphics, framing and coherence and cohesiveness of the game system [41]. As serious games are often used for education, Freire et al. [22] discuss how knowledge from game analytics and learning analytics can be combined in order to improve understanding of educational serious games.

A specific form of game research entails the analysis of gameplay videos, often referred to as Let's Play-videos in the literature. Radde-Antweiler et al. define these videos as "self-recorded gaming videos in which the respective gamers, the 'Let's Players', comment on their journey through the game as well as on various aspects of it" [47]. These videos are often available online on platforms such as YouTube and Twitch. Once again, the purposes of this type of research can vary considerably. Milam and El Nasr [40] used gameplay videos of 21 games to analyze their level designs and how these are used to guide the player through the game. Mun et al. [44] introduced a framework to evaluate the temporal reasoning capabilities of algorithms by using those algorithms to answer questions based on segments of gameplay videos. Marczak et al. [36] used image processing techniques to extract quantitative data from gameplay videos they recorded, which could be combined with psychophysiological responses or self-report metrics to measure how the player experiences a game. A method to perform event and highlight detection on Twitch streams of League of Legends was developed by Chu and Chou [16]. To do so, they used techniques from text recognition on the content on the screen and in the chat boxes paired with the stream and also adapted regular video event detection techniques to the game domain. While there has been an increasing amount of published game research over the past years, there are still few articles that discuss visual and semantic analysis of gameplay videos such as those by Marckzak et al. [36] and Chu and Chou [16]. The topic of this thesis is visual semantic analysis of gameplay videos, specifically of the game 'This War of Mine'. It is part of an in-progress research project by De Smale et al. [20].

'This War of Mine' [4, 5] is a video game where the player plays as a group of civilians trying to survive during the Bosnian War. The gameplay consists primarily of two phases, a day phase and a night phase. During the day, the characters hide in a shelter, because it is too dangerous to go outside. In this shelter, the characters can build items to improve the shelter or make it more comfortable to live there, eat, sleep and so on. During the night the player can send out a character to go scavenging in the city to find resources, such as water, food, medicine and materials to construct items. For example wood and electrical parts to make a stove or bed. Scavenging can be dangerous due to soldiers being out there at night, who will attack the character. There can also be encounters with less peaceful other scavenging civilians. The player has to make moral decisions to keep his characters alive. One such decision is when the player goes scavenging at a supermarket. There a conversation can be overheard and an event can be watched through a keyhole, where an armed drunk soldier tries to rape a scavenging woman. The player then has the choice to remain passive and let the rape take place, which will allow the character to safely loot the supermarket. However, this also negatively affects the character's emotional state, because they witnessed a rape and did not try to stop it. Alternatively, the player can interfere by attacking the soldier. This comes with the risk of being killed, because the soldier is armed. However, the reward for intervening is bigger, because the dead soldier's equipment can be looted, which will make protecting the shelter from raiders easier and also gives protection against other hostile characters when scavenging further on in the game.

Hundreds of gameplay videos of 'This War of Mine' are available online through YouTube. These videos can be used to analyze the game and the choices its players make in situations such as the one described above. They can be analyzed manually, but the topic of this thesis is investigating to what degree this process can be automated, using computer vision techniques to analyze gameplay videos. The case study discussed here is about analyzing the scene in the supermarket. Doing this analysis consists of three parts:

1. Detecting whether the supermarket occurs in a video
2. Detecting whether the scene with the soldier takes place if the supermarket is visited
3. Determining what choice the player makes if this scene takes place

239 videos of 'This War of Mine' have been studied, which contain 175 occurrences of the supermarket in 139 videos. A more thorough description of the dataset is given in Subsection 4.1

In section 2 a motivation is given for investigating specifically the scene in the supermarket as well as a motivation for why this topic is interesting and relevant to look at from a technical perspective. Section 3 describes related work about techniques used in the literature about video analysis. The formal research questions for this thesis are presented in Section 4. The approaches used to answer these research questions are discussed in Section 5. The experiments performed to test the developed approaches are discussed in Section 6 followed by a discussion about the results in Section 7. Lastly, conclusions are drawn and formal answers are given to the research questions in Section 8.

2 Motivation

The research presented in this thesis is part of an ongoing research project by De Smale et al. [20]. The goal of this project is to visualize the decision paths a player makes when playing 'This War of Mine' through image analysis, using computer vision techniques. This specific game was chosen as it is "potentially valuable for peace education and conflict resolution" [20]. The ultimate goal is to build a decision tree of the behavioral choices made by the player when playing the game.

Extracting the decision path of a player playing the game has been split into two separate problems. The first is to recognize which characters the player decides to use to go scavenging, along with which characters were available. After all, if there is only one character available, this choice is less meaningful. It then boils down to taking the risk of going scavenging or staying in the shelter, but dying to dehydration or starvation. Also part of this problem is recognizing which items the player chooses to take with him or her when going scavenging and what items are brought back to the shelter in such a scavenging run. This sub-problem is not part of this research and is concurrently being worked on independently.

The second part of visualizing the player's decision path is to extract the choices the player makes in the moral dilemmas presented to him or her. The rape scene in the supermarket is an example of such a dilemma and the case study chosen for this thesis. The primary goal of this thesis is therefore to find out to which degree recognizing the player's choices can be automated. The secondary goal is to investigate, if it does turn out to be possible to get good results in an automated way, how well the developed method can be generalized to other dilemmas in 'This War of Mine' as well as to entirely different games.

Semantically analyzing player decisions in gameplay videos is also a relevant challenge from a technical point of view. Little research on this specific topic has been published. Even when expanding the search to the broader topic of video analysis, the focus is often on classifying videos or segments or analyzing their structure [24, 15, 12, 43], not on semantically analyzing the events that take place in the video. Video footage is also combined with other types of information such as sound or captions accompanying YouTube videos [56, 14] to perform semantic analysis, whereas in this research the focus is on using only video footage. In cases where the actual events taking place in the video are analyzed, the domains investigated are often very specific and lots of domain knowledge can be used in them. Examples are analysis of sports videos [52, 25, 59, 62], surveillance videos [32] and types of movement [51].

The topic of this study is relevant, because it does not require access to the source code of the game being investigated, nor does a game have to be developed by researchers. If commercial games can be more easily used for research, scientists outside technical fields such as computer science could use these games for case studies. For example social studies could use the technique being developed here, instead of having to manually analyze all videos in their case study, speeding up the process. It could also save money, because existing games might be used to investigate topics such as peace education and conflict resolution (the reason 'This War of Mine' was selected in this research project). No new game would have to be developed, saving considerable time and money.

3 Related work

This section gives an overview of techniques and methods that are used in various computer vision related tasks, primarily focusing on analyzing video footage. It is by no means an exhaustive list as that would be beyond the scope of this research and should therefore not be interpreted as such.

Simple, but often-used features are color-related features. The literature shows a great variety of uses for these relatively simple and computationally inexpensive features. Marckzak et al. [36] used color ratios to detect objects of interest such as health bars and color schemes to detect menu screens. Mühling et al. [43] used color moments to detect an inactive game state in a first-person shooter game. Dominant color regions were used in several articles regarding automatic analysis of sports videos to detect the field or specific objects such as the referee [25, 59, 62, 21]. Color correlograms were used, among other features, to represent shots in an algorithm to detect scenes in a video [42]. The color features that appear to have been used the most in the literature are color histograms.

These have been used for purposes from detecting special effects in Twitch streams [16] to taxonomic classification of online videos [56]. They are also often used for detection and segmentation of scenes or segments in videos [42, 58, 18, 7, 61, 62, 13].

Different features that are often used, yet are also still relatively simple are edge-related features. Edge-related features have been used for purposes such as text detection [16], detection and segmentation of scenes and segments in video footage [62, 42, 45, 25, 29, 13], as one of the features in taxonomic video classification [56], for line and region detection [25] and even to detect aliasing in shadow-only images used for shadow map artifact detection [45].

More complex features originating in image processing have also been applied to video processing. Techniques such as Scale-Invariant Feature Transform (SIFT) [35], which finds a set of descriptive keypoints (features) of an image that are "largely invariant to changes in scale, illumination and local affine distortions" [35]. While the original article used SIFT to detect objects in images [35], it can also be used to extract information from videos. SIFT features have been used to create a bag-of-visual words through K-means clustering, which was used to train a SVM to estimate camera viewpoints in soccer videos [52]. A similar approach was used by Hentschel et al. [24] to classify and annotate segments of lecture videos. Mei et al. [39] used SIFT-features for near-duplicate keyframe detection. An alternative to SIFT are Speeded-Up Robust Features (SURF), which achieve similar results, while being computationally cheaper [9]. It has also been used in the video processing domain. Apostolidis and Mezaris [7] achieved an even larger speedup by porting SURF to the GPU in their system to detect shot boundaries in video footage. Both SIFT and SURF are implemented in OpenCV as well as several other feature descriptors that also see use in the literature. Examples are Fast Retina Keypoint (FREAK) [6], Binary Robust Invariant Scalable Keypoints (BRISK) [34], Features from Accelerated Segment Test (FAST) [49], Binary Robust Independent Elementary Features (BRIEF) [10] and Oriented FAST and Rotated BRIEF (ORB) [50].

Processing videos means that types of features can be used that cannot be used during regular image processing, which consists of individual images with no temporal relationships. Working with video footage makes it possible to detect and track motion, which can give valuable information in various domains. The literature shows many methods to detect motion from videos. Examples are taking the sum of absolute differences between horizontal and vertical projections of successive frames [25], deformable part models [52], motion vector models/fields [21, 61, 62, 13, 19], optical flow [16], moving edge maps combined with connected components analysis [29] and a continuous hidden Markov model to detect human postures in surveillance videos [32].

Videos contain a time dimension, which inherently means there are temporal relationships in the video, not limited to motion alone. These temporal relationships have been used as part of video analysis in the literature. Mun et al. [44] developed a dataset to test video analysis algorithms on their temporal reasoning capabilities. They also built a 3D fully convolutional neural network to detect temporal relationships [44]. Mohanta et al. [42] used the time of appearance of visually similar frames to improve scene detection. A similar approach was used by Yeung et al. [60]. Huang et al. [25] introduced a temporal intervening network as an extension to a dynamic Bayesian network, which they used for semantic analysis of soccer videos. Essentially this temporal intervening network was a set of additional rules, based on temporal relationships, for the Bayesian network. Gu et al. [23] used an energy minimization method to segment videos into scenes. In their model context energy denoted the temporal relationships between shots, where a shot contained more energy relative to another shot if these shots were assigned to different scenes while being temporally close to each other as well as visually similar [23].

Earlier in this section various features for video processing have been discussed. The methods used to train a system or part of a system with these features are also plentiful. Hidden Markov Models are sometimes used [32, 17] and so are regular Bayesian networks [12, 25]. Another sometimes used probabilistic model is the Gaussian mixture model [32, 63, 17]. Neural networks have been used in the past [59], but have picked up in popularity in recent years [44, 18, 37]. The most encountered method in the literature study for this research are support vector machines (SVMs). However, as noted before, this literature study is by no means a complete overview of the field and SVMs are therefore not necessarily the most used method in the entire field. Additionally, the majority of the articles studied here, which used SVMs are at least five years old [24, 43, 21, 53, 55, 54, 39]. When looking only at recent articles, an equal emount of articles that use SVMs [16, 52, 17] are found to those that use neural networks [44, 18, 37]. The advantage of SVMs over neural networks lies in their interpretability. While SVMs are by no means an easy-to-interpret method, especially when not using linear SVMs, they are not a completely black box method like neural networks. When training a SVM with a bag-of-visual-words model such as in the approaches taken by Sharma et al. [52] and Hentschel et al. [24] the words (features) on the decision boundary of the SVM can be inspected to gain some insights into why those features are descriptive, which is generally not possible with neural networks. Also, the current trend seems to be that neural networks are used with large amounts of data, so-called big data, whereas SVMs can also be used when only a smaller dataset is available.

The advantage of neural networks, in the case of image processing usually convolutional neural networks, is that they can often achieve better performance than SVMs if enough data is available. However, even when only a smaller dataset is available convolutional neural networks can still yield satisfactory results through the use of transfer learning. The simplest form of transfer learning on convolutional neural networks is to copy an entire existing network's architecture and all its weights, only replacing its last layer with one to classify to the classes of the new problem. More complex forms of transfer learning are when only a subset of the layers is copied or when weights of certain layers are frozen, i.e. these weights cannot be changed while retraining the network on a new problem. The benefit of transfer learning is that it can save considerable time compared to building a new domain-specific network from scratch. Unfortunately there is not always a suitable network available for transfer learning. The effectiveness and applicability of transfer learning have to be investigated on a case-by-case basis. However, when transfer learning is possible it can even be used to achieve state-of-the-art results as was demonstrated by Oquab et al. [46].

4 Research questions

In this section a description of the dataset used during this research will be given. This dataset will be used to answer the research questions posed later in this section. The dataset is described in Subsection 4.1. As noted in the introduction, this research consists of three parts, each part having its own research question. Detecting whether the supermarket occurs in a video is discussed in Subsection 4.2. Detecting whether the scene with the soldier takes place if the supermarket is visited is discussed in Subsection 4.3. Lastly, determining what choice the player makes if the scene with the soldier takes place is discussed in Subsection 4.4.

4.1 Description of dataset

This study is part of ongoing research project. Earlier in this project, a set of videos of 'This War of Mine' had been scraped off the internet to be used. This was the dataset referred to by De Smale et al. [20]. In this article 500 videos are mentioned. However, in a scenario specific search only 314 videos remained. A sizable part of these videos have since been removed from the internet (or at least they can no longer be found using the YouTube video identifier that was recorded for the original dataset). Some of the identifiers linked to videos of other games or other topics entirely. All identifiers which did not link to a video of 'This War of Mine' have been removed for this study. The reason for removing the videos that did not contain 'This War of Mine' are twofold. First, when analyzing videos of 'This War of Mine', it is a reasonable assumption that the videos indeed contain footage of 'This War of Mine'. Second, this study uses YouTube videos, where the title can make clear whether the video contains footage of 'This War of Mine'. All but one video in the dataset used for this study had 'This War of Mine' in its title (or variations where the capital letters differed). The only one that did not contain it, contained its abbreviation 'TWOM'. Therefore, simple text analysis on the video's title should suffice to determine whether a video contains footage of 'This War of Mine', which allows the assumption that all videos used for this study contain footage of the game. This assumption should ease the video processing problem in this study. Furthermore, some videos were of very poor quality. They were for example extremely dark, which would make it difficult to recognize anything in them. In one video the screen also turned black multiple times outside of loading screens. Another video was a review video of the game, where only parts of the screen contained footage of the game and the rest of the screen contained a background and the video's host. These poor quality videos have been filtered out of the dataset. After removing deleted and unrelated videos and videos of very poor quality from the dataset 239 videos remained. Some of these videos contained multiple occurrences of the supermarket. These videos have been split in the dataset used here, so that each entry refers to one trip to the supermarket or does not contain the supermarket at all (the negative samples). Out of the 239 videos used here, 139 videos contained the supermarket for a total of 175 unique supermarket visits (entries in the dataset).

The videos in this dataset vary in resolution from 360p to 1440p. Certain videos also contain an introduction and or ending segment of the YouTube channel they were published on. These segments do not contain footage of 'This War of Mine' and can have completely different visual content than the actual gameplay footage. Videos can also have parts of the screen occluded by a display of the streamer playing the game. These are usually windows in a corner of the screen where a recording of the streamer is displayed. This can make parts of the user interface or the visited locations impossible to see.

For this study each video in the dataset has been annotated. The annotations differ by the contents of the entry. Each entry contains a boolean that represents whether it contains the supermarket. Each entry also contains a note where peculiarities of the video are described. For example when a glitch takes place. In the majority of the entries, the note is empty. If and only if an entry contains the supermarket, it also contains a boolean that represents whether the scene with the soldier takes place and a start and end time. The start and end times denote the points in time, in the video tied to the entry, where the supermarket appears and disappears. This time is in the [minute:second] format. The entries that contain the scene with the supermarket also contain a start and end time for this scene, once again in the [minute:second] format. These entries also contain annotations about the choice the player made. Here there are three possibilities:

1. Intervene: the player stops or attempts to stop the soldier from raping the woman. Once this choice is the made, the woman will escape, but it is possible for the player's character to be killed by the soldier. The scene is defined to have ended when the soldier has been killed, the player's character has been killed, the soldier retreats when he is very low on health or the player's character has escaped the supermarket. The note described above is used when the player's character has been killed, the soldier retreats or the player escaped. These events occur only in a minority of the cases. The vast majority of cases sees the player's character kill the soldier.

2. Passive: the player does not try to stop the soldier. The scene is defined to have ended when the soldier and the woman enter a shack at the edge of the map, which the soldier locks before the actual rape takes place.

3. Kill both: this is a rare occurrence where the player kills both the soldier and the woman. The scene is defined to have ended when they have both been killed.

There are 55 occurrences of the scene with the soldier. In 38 of those the player chooses to intervene, in 15 cases the player remains passive and in only 2 cases does the player kill both the soldier and the woman. Of these two cases, in one this happens by accident due to the player clicking on the wrong button and in the other it is a player-defined challenge to kill all encountered non-playable characters (NPC's).

4.2 Research question 1

The first problem that needs to be solved is detecting whether a video contains the supermarket. The other two problems are irrelevant if this is not the case. Therefore successfully solving this problem is a requirement for solving the other two problems. Given the dataset described above, the assumption is made that the video being processed does indeed contain gameplay footage of 'This War of Mine'. This leads to the following formal research question:
How can it be determined if a video containing gameplay footage of 'This War of Mine' contains a scene that takes place at the supermarket location, which is unique within the borders of the game?

To answer the other two research questions, it is not enough to only know if a gameplay video of 'This War of Mine' contains a scene in the supermarket, but also when a visit to the supermarket takes place, when it begins and ends. Therefore the first research question can be divided into two more concrete sub-questions:

1. How can it be determined if a frame containing gameplay footage of 'This War of Mine' displays the supermarket location, which is unique within the borders of the game?

2. How can it be determined when a visit to the supermarket, which is unique within the borders of the game, in gameplay footage of 'This War of Mine' begins and ends?

4.3 Research question 2

The second problem to be solved is detecting whether the scene with the soldier takes place when the supermarket is visited. This problem uses the assumption that it is known that the supermarket is visited and therefore requires the first research question to

be answered before this problem can be solved. When the supermarket is visited, there are four possibilities. The first time the supermarket is visited, either the scene with the soldier will take place or a scene with three other scavengers, which are armed with guns, takes place. The third possibility is when the supermarket is visited a second time (and any times after that) after the scene with the soldier has taken place. Here, the player will encounter a woman who comments on the event with the soldier that has taken place. Her text depends on the choice the player has made in the scene with the soldier. The fourth and last possibility is that the supermarket is empty. In this case, only the player's character is present at the location. This scenario occurs on a second and any subsequent visits if during the first visit the scene with the three other scavengers occurred. This scenario will also take place if during the player's first visit the scene with the soldier took place and the player has killed the woman that comments on this event during the second or any subsequent visits. Any visits after this woman has been killed will also show an empty supermarket, where the player's character is the only character present in the supermarket. For the purposes of this study this problem is seen as a binary problem. The scene with the soldier either takes place or it does not. The three scenarios where the scene does not take place are considered as one case, the case where the scene with the soldier does not take place. The formal second research question then becomes:
How can it be determined whether a scene takes place in the supermarket location in 'This War of Mine' where a soldier attempts to rape a woman when it is known that the gameplay footage being studied contains a scene in the supermarket, which is unique within the borders of the game?

4.4 Research question 3

The third and last problem to be solved is detecting what choice the player makes when the scene with the soldier takes place. This problem uses two assumptions, that the video being studied contains footage of 'This War of Mine' and that in this footage the scene with the soldier takes place. Therefore, it requires the first two research questions to be answered before this question can be answered. There are three possibilities to the player: to intervene, to remain passive and to kill both the soldier and the woman. What should be noted is that choosing to kill both the soldier and the woman is essentially a special case of intervening, because the player will first intervene in the scene and then proceed to kill both these characters. In the dataset this choice is only made twice, which will make it hard to uniquely distinguish. Nonetheless, it will be considered as a separate case. To what degree it is possible to uniquely distinguish this case will become apparent during this research and it may be dropped as a unique case if it cannot be reliably detected. The third formal research question then becomes:
How can it be determined whether the player chooses to intervene, remain passive or kill all characters in a video displaying a scene from 'This War of Mine', where a soldier attempts to rape a woman in a supermarket, which is unique within the borders of the game?

5 Approach

In this section the preprocessing of the dataset is first discussed before introducing the used approaches to solve each of the three sub-problems and to answer their corresponding research questions. The preprocessing of the dataset is discussed in Subsection 5.1. The used approach to solve detecting the supermarket is presented in Subsection 5.2. The method used to detect the scene with the soldier when it is known that the supermarket is visited in a video is discussed in Subsection 5.3 and the approach used to detect the player's choice in this scene is discussed in Subsection 5.4. All the operations discussed in this section were performed using OpenCV in C++ unless stated otherwise.

5.1 Preprocessing

The gameplay footage containing the supermarket does not contain only the supermarket. There are also multiple overlay screens that can cover the screen displaying the supermarket. When an overlay screen appears, the content behind it is covered. The parts of the screen that are not part of the overlay screen are blurred, making the location impossible to recognize. The most common overlay is the inventory or backpack screen. This screen appears when the player wants to collect loot from piles in a location or when the player clicks the backpack button to check what items he or she has with him or her. There are also some objects in locations that can be inspected. These objects are represented by a looking glass on the map. When such a looking glass is clicked on, an overlay with text appears that gives more information about the object. Usually the text displayed here is for lore-building to add atmosphere to the game, but sometimes it can contain hints to discover hidden loot locations (although the latter is not the case in the supermarket). Other overlays that can appear are the menu screen, the pause screen and the character biography screen, which gives some background information about the selected character and their views on events that have taken place in the game so far. These views represent the emotional state of the character. What all these possible overlays have in common is that they obscure the location being visited. When detecting features which can be used to train a model that can uniquely identify a location, the supermarket in this study, these overlays should not be taken into account as they would reduce the quality of the results. Therefore these overlays have to be recognized so they can be filtered out of the footage.

When the overlays appear on the screen, the content of the screen changes significantly. This fact can be used to detect these overlays. By calculating three one-dimensional color histograms, for the red, green and blue channels individually, and comparing the histograms between frames, changes can be detected. To compare histograms two built-in histogram comparison methods from OpenCV were used. The histogram correlation method was used as well as the Chi-squared method. Two methods were used to improve the accuracy of the results. When using only one method the number of false negatives was too high. All histogram comparison methods from OpenCV were tested and these two gave the best results. The correlation method in OpenCV is defined as: "

$$d(H_1,H_2) = \frac{\sum_I \left(H_1(I) - \overline{H}_1\right)\left(H_2(I) - \overline{H}_2\right)}{\sqrt{\sum_I \left(H_1(I) - \overline{H}_1\right)^2 \sum_I \left(H_2(I) - \overline{H}_2\right)^2}} \tag{1}$$

where

$$\overline{H}_k = \frac{1}{N}\sum_J H_k(J) \tag{2}$$

and N is a total number of histogram bins." [2]. The Chi-square method is defined in OpenCV as: "

$$d(H_1,H_2) = \sum_I \frac{\left(H_1(I) - H_2(I)\right)^2}{H_1(I)} \tag{3}$$

" [2]. Both of these methods return a number which is compared against a threshold, where each method has its own threshold. When either of the thresholds is exceeded a screen change is detected.

Color histograms are simple and fast-to-compute features. However, comparing all frames turned out to be too slow due to the sheer volume of data. Additionally, comparing all frames caused false positives due to screen fading. When the location appears or disappears, it fades in from or out to the loading screen, which causes false positives. In some videos screen fading also occurred when an overlay appeared or disappeared, once again causing false positives. To solve this problem and lower the computational load, only each 30$^{\text{th}}$ frame of a scene containing the supermarket is processed. The start and end times of the scene are taken from the annotations made earlier and described in Subsection 4.1. A new problem arose from only looking at every 30$^{\text{th}}$ frame. Between two 30$^{\text{th}}$ frames the player can have moved considerably in the location, which causes the camera viewpoint to change location as well. These camera changes cause differences in the screen content above the threshold to detect a screen change, while no screen overlay has appeared. The solution to this problem was to check the entire interval of 30 frames when a screen change was detected between the two 30$^{\text{th}}$ frames. Between successive frames the camera movement is smaller and therefore the changes in histograms are below the screen change detection threshold. Only when the threshold is exceeded between two successive frames in the 30 frame interval, a screen change is definitively detected.

This method was used to find the frame numbers where a screen change takes place, if those frames are part of a scene that contains the supermarket (the annotations of the dataset were used for this). Only every 30$^{\text{th}}$ frame was considered. Some screen changes were still missed or erroneously detected by this method. These were usually in particularly difficult cases, such as a sudden flash on the screen, either due to the recording equipment or a lightning strike in the game. Sudden large camera jumps caused false positives. It is possible in the game to move the camera without moving the character. This way the location can be quickly explored without showing details such as objects that can be looted. However it is also possible to then quickly move the view back to the character through a jump in the camera viewpoint, which is detected as a screen change while no overlay appears. Because this does not happen gradually, checking the entire interval does not prevent these camera jumps from causing false positives. However, automatically detecting rare events such as these would cost disproportionately large amounts of work. As there were no ground truths available for correct frame numbers of appearances of overlays, the results were manually verified and if necessary corrected. The only remaining possibility for not detecting an overlay now is when an overlay both appears and disappears within a 30 frame interval. This does not matter, because the frames in that interval will not be used for further processing and therefore any such missed overlay appearances and disappearances within a 30 frame interval will not affect results of later processing steps.

The preprocessing described above results in a list of frame numbers where the supermarket appears or disappears. Each entry in the dataset has its own list if that entry contains a visit to the supermarket. These lists also include changes caused by screen fading at the start or end of the segment containing the supermarket. All other changes in the list represent an overlay screen appearing or disappearing. Additionally for each entry in the dataset a boolean has been added that represents whether the first frame of the scene clearly contains the supermarket. If it does, this boolean starting value is true, otherwise it is false. By keeping track of the frame number of the frame being processed during further processing, this boolean can be used to determine whether the frame should be processed. The value of this boolean is flipped around whenever a screen change has occurred at the current frame number according to the list of screen changes discussed above. If after this check the boolean is true, it means the frame contains the supermarket and should be processed. If it is false, it means the supermarket is not clearly visible and therefore the frame should not be processed.

5.2 Detecting the supermarket

Detecting the supermarket can be split into two more concrete sub-problems, as discussed in Subsection 4.2. The first sub-problem is determining whether a frame displays the supermarket, which is discussed in Subsubsection 5.2.1. The second sub-problem is to determine when a visit to the supermarket begins and ends, which is discussed in Subsubsection 5.2.2.

5.2.1 Determining if a frame displays the supermarket

To recognize the supermarket, keypoints were extracted from every 30th frame in 50 segments containing the supermarket. One video can contain multiple segments in the supermarket. In this case, each visit has its own entry and annotations in the dataset. Frames not containing the supermarket, overlay screens or screen fading during a visit to the supermarket are excluded here. Additionally keypoints were extracted from every 300th frame in 50 videos not containing the supermarket. The keypoints are extracted using SIFT [35] and descriptors are also computed using SIFT. Recently, some studies have compared multiple feature descriptors and they concluded that the best results were still accomplished using SIFT [27, 28] compared to more recent methods. This, combined with there being no requirement for the program to process videos in real-time, was the reason SIFT was chosen over other feature descriptors. The SIFT descriptors were clustered using K-means clustering to construct a bag-of-visual-words. This process yields a codebook, also referred to as a dictionary in the literature. Separate codebooks were constructed for the descriptors extracted from footage that contains the supermarket and footage that does not. Frames from other videos can then be compared against these two codebooks, which will yield two matrices, representing bag-of-visual-words histograms. These can be concatenated to form the features to train or test a binary SVM classifier. Once trained, this SVM predicts whether a given frame contains the supermarket. A 2-class SVM with C-Support Vector Classification and a linear kernel was trained on 50 different segments containing the supermarket than the segments used to create the codebooks. Every 30th frame was processed, but once again frames not containing the supermarket, overlay screens and frames containing screen fading were excluded. The negative training samples consisted of every 60th frame for 30 videos not containing the supermarket, that were not used to create the codebooks. The SVM was trained using OpenCV's trainAuto method, which automatically optimizes the parameters of the SVM and applies 10-fold cross-validation.

The features calculated to train the SVM can also be directly used to train a K-nearest neighbor classifier, so this was done to compare the results of the two classification methods. To set K the rule of thumb that K should equal the square root of the number of training samples was used. The SVM performed better, which will be shown in Subsubsection 6.1.2. All other results referred to in this Subsubsection will also be presented there. Since the SVM performed better, further processing was only performed on the classifications made by the SVM.

To improve the results of the SVM classification, features were computed for the average of every interval of 30 frames instead of only looking at every 30th frame. The idea behind this was that the average of the interval would better capture the information contained within that frame interval than only one frame at a fixed location. However, using the average of frame intervals led to worse results so this method was discarded.

A different method to improve the classification results was added, a 'temporal smoothing' method. This method looks at the predicted label of each classified frame and compares it to the predicted labels of the surrounding frames. If all surrounding frames have a different predicted label than the current frame, the current frame is likely to have been classified incorrectly. For example if in a video not containing the supermarket, with a frame rate of 30 frames per second , the frames 1, 31, ... , 121 and the frames 181, 211, ... , 301 are all predicted as not containing the supermarket, but frame 151 is predicted to contain the supermarket, this is probably a false positive, because all the surrounding frames do not contain the supermarket and from watching videos in the dataset and through playing 'This War of Mine', the assumption can be made that a visit to the supermarket lasts as least 10 seconds. I.e. a one second visit cannot take place and would therefore have to be a false positive. The number of surrounding frames that are looked at for each classified frame can be changed. All window sizes from looking at one classified frame before and after each frame up to and including 5 classified frames before and after each frame were tested. The full results are presented in Subsubsection 6.1.2. The best results on the training set and primary test set were obtained with a window size of 2, where only the first classified frames before and after each frame were used for 'temporal smoothing'. This means that the closest surrounding classified frames contain the most information about the current frame, which is not surprising, but also that adding information about more surrounding frames does not lead to larger improvements, which is not immediately apparent. On some alternative smaller test sets a larger window size led to slightly better results, but because on the largest datasets a window of size 2 led to the biggest improvements, this window size was selected for further processing. An additional note that needs to be made is that the overall improvement of the results is not large, at most a few percent, but compared to generating the features to classify frames using the SVM, it is a fast method, that does consistently lead to a small improvement in the results. Therefore, the method was deemed worth using, despite only yielding a small improvement of the results.

5.2.2 Determining the beginning and ending of a visit to the supermarket

The SVM described in Subsubsection 5.2.1 only classifies individual frames. However, because a video can contain multiple visits to the supermarket it is necessary to determine at what frames a visit begins and ends. Otherwise solving the other two research problems would become more difficult and may also lead to incorrect results. For example if in a second visit strong evidence is found for an empty supermarket (i.e. no scene with the soldier), this may erroneously classify the entire video as a scene in an empty supermarket even if the scene with the soldier has taken place earlier.

To determine where a supermarket visit starts and ends the predictions of the SVM for individual frames are used. These predictions are ordered based on the frame they correspond to. If a frame has been predicted as positive and no scene has started yet (i.e. as containing the supermarket), a window of 900 frames (30 seconds) after that frame is investigated. Every classified frame in that window (every 30$^{\text{th}}$ frame) is looked at and the number of frames in that window that are also classified as positive are counted. If there are less than 900 frames remaining in the video, every 30$^{\text{th}}$ frame until the end of the video is taken into account. If the percentage of positively classified frames in this window exceeds a threshold, it is determined that a scene starts at the frame before the window. The reason for this window is that supermarket visits have a certain minimum length, but only looking at consecutively positively classified frames does not work well, because overlay screens can appear during a supermarket visit. The threshold was set at

0.4. This value was chosen based on the lengths of the supermarket visits in the videos in the dataset and on experience from playing the game.

Determining the end of a visit is done in a similar way. The same 900 frame window is investigated, but here it is checked if none of the frames are classified as positive. However, a 900 frame interval of negative predictions does not necessarily mean that a scene has ended. It could also be an overlay screen and that the scene will continue after this overlay screen disappears. Therefore, if no positive prediction is found in the 900 frame window, the 1800 frames after that are investigated as well for a total of a 2700 frame window (90 seconds). If the number of positive frames in this 2700 frame window is smaller than or equal to a threshold of $\frac{1}{30}$, the scene is determined to have ended. The 2700 frame window was selected based on the smallest time between scenes in the dataset and on experience of how quickly nightly raids can succeed one another gained from playing the game. There was one video in the complete dataset where two visits directly succeeded each other, but this is an outlier and was also not a proper Let's Play video, but a compilation of ways to kill the soldier in the supermarket.

While this method worked well to identify supermarket visits, it also detected many visits erroneously. This was caused by the false positives of the frame predictions often being clustered. To combat this, initially a second SVM was trained on the false positives against the true positives of the frame classification method described in Subsubsection 5.2.1 on the extended training set. The idea behind this was that it could distinguish between true displays of the supermarket and other scenes. Unfortunately this method did not work well. The frame classification method used an SVM with a linear kernel, so instead SVMs with different kernels were trained on the regular and extended training sets and compared with the existing SVM. Of these alternative SVMs, the best results were obtained using an SVM with a Radial Basis Function (RBF) kernel on the extended training set. While this SVM achieved lower recall than the SVM with a linear kernel, it also achieved a higher precision and similar accuracies and F1-scores. Using this SVM with a RBF kernel, fewer visits were erroneously detected. The full results of the SVM with the RBF kernel on various datasets are presented in Subsubsection 6.1.2 and a comparison of the scenes detected by the method described in this section between the SVM with a linear kernel and a RBF kernel can be found in Subsubsection 6.1.3. The SVM with an RBF kernel performs equally or better than the linear SVM on the scene times training set. It does not detect any scenes incorrectly there. On the scene times test set there are differences between the methods. The linear SVM detects two more scene starts correctly and one more scene end correctly than the SVM with the RBF kernel. However, it also detects 15 more scenes incorrectly. So overall, the SVM with a RBF kernel performed better on the scene times test set than the linear SVM. Of the scene ends identified too late in the scene times test set in several cases this was due to an overlay screen being present for a prolonged period of time near the end of a supermarket visit. This is due a strategy in the game of moving all loot in a location to one container and then picking the best items from that container just before leaving the location. In these cases an overlay screen is present for more than 2700 frames, which causes the method described in this section to detect the appearance of such an overlay screen to be the end of a visit. While this is not correct, these wrongly predicted scene ends do not affect the detection of the scene with the soldier, because after the disappearance of such an overlay screen, the character only moves to the exit of the location, i.e. the videos do not contain any information anymore about whether the scene with the soldier occurs. In an attempt to improve the results different methods for detecting keypoints and extracting descriptors were used, ORB and SURF, but these led to worse results. Using these methods individually performed worse than SIFT. Combining them with the existing SIFT features actually decreased the quality of the results. Therefore the choice was made, also due to time constraints, to use the existing method of SIFT features on a SVM with a RBF kernel and move on to solving the second research problem. The SVM with a RBF kernel was chosen over the linear SVM due to strictly better results on the training set and overall better results on the test set.

One last thing that can be done to make the detection of starts and ends of supermarket visits slightly more accurate in certain videos is to detect the occurrence of loading screens before and after the visit. In a significant portion of the videos containing the supermarket a loading screen is displayed before the start or after the end of a visit. This screen is almost entirely black, containing only some white text. These screens can be detected by converting frames to grayscale and calculating the percentage of black pixels on the screen. To this end, nearly black pixels are thresholded to fully black pixels. A threshold of grayscale value 3 was used. If at least 95% of the pixels on the screen are black, it is determined that a loading screen is displayed. By looking at the 300 frames before the start of a supermarket visit or after its end, detected by the method described above, and when a loading screen is displayed, the start and end frames of that visit can be determined more precisely. A 300 frame window was selected to accommodate for frames classified incorrectly due to screen fading. Not all videos contain loading screens. If this method does not detect a loading screen the frames that are definitively considered to be the start or end of a scene are the frames that are found by the method described earlier in this Subsubsection.

5.3 Detecting the scene with the soldier

Detecting whether the scene with the soldier takes place during a supermarket visit can be split into two more concrete sub-problems, in the same vein as detecting whether the supermarket is visited in a video. The first sub-problem is determining whether a frame displays the scene with the soldier. This is discussed in Subsubsection 5.3.1. The second sub-problem is to determine whether a supermarket visit does or does not contain the scene with the soldier, which corresponds to the second research question. This is discussed in Subsubsection 5.3.2.

5.3.1 Determining if a frame displays the scene with the soldier

To detect the scene with the soldier, initially the same method as to detect the supermarket was used. The idea behind this was that the approach could be used in an iterative fashion on a more specific dataset. Therefore, keypoints were extracted from every 30th frame in 18 videos containing the scene and 40 videos not containing the scene. Frames containing overlay screens or screen fading were filtered out in the 18 videos containing the scene with the soldier. The class imbalance above is due to there only being 55 videos that contain the scene in the dataset, whereas there are 120 videos not containing the scene. SIFT was used to extract the keypoints and also to calculate the descriptors from these keypoints. The SIFT descriptors were then clustered, using K-means clustering, into two codebooks, one for videos with and one for videos without the scene. Since both classes have their own codebook, the impact of the difference in number of videos per class used to extract features from, is strongly reduced, because there are enough samples per class to generate a codebook.

The codebooks were used to create bag-of-visual-words histograms on the training and test sets. The training set consisted of 18 videos containing the scene where every 30th frame containing the scene was processed, and of 40 videos not containing the scene where every 30th frame was processed. The test set consisted of 19 videos containing the scene, where once again every 30th frame was processed, and 40 videos not containing the scene, where also every 30th frame was processed. In both the training set and test set overlay screens were filtered out in the positive samples, but not in the negative samples. The bag-of-visual-words histograms were used as features to train SVMs, with a linear and a RBF kernel. The full results are displayed in Subsection 6.2.2. Unfortunately, both of these SVMs performed poorly. Upon investigating the detected keypoints and detectors, it turned out that a significant part of the detected keypoints were part of the user interface (UI), which does not contain any information about which scene takes place. To prevent the UI from influencing the results, new codebooks were generated based on frames where the UI had been filtered out through an image mask. These new codebooks were then used to generate new bag-of-visual-words histograms on the training and test sets, but this time, the UI had been filtered out of all frames here as well. New SVMs were trained, using these modified bag-of-visual-words histograms, now also testing a SVM with a histogram intersection kernel. Filtering out the UI did lead to an improvement in the results, but overall the results were not yet satisfactory with a highest recall of 64.9%.

Several different keypoint descriptors were tried to see if they would lead to an improvement in the results. These descriptors were SURF, ORB, BRISK and KAZE, but they did not lead to improvements. So far, the keypoints have been detected on frames directly. In an attempt to only detect keypoints on clear objects and not also in the background, keypoints were detected on Canny edge maps. However, bag-of-visual-words histograms extracted from these edge maps led to SVMs that always predicted the same value. The bag-of-visual-words histograms of Canny edge maps turned out to not be good enough features to distinguish between classes. Therefore, this method was discarded.

The objects in frames that give the most information about which scene is taking place, are the characters, i.e. the presence of the soldier and the woman, the scavengers or no characters other than the player's character. The non-playable characters also appear at approximately the same locations, usually near the middle of the screen. One of the disadvantages of the bag-of-visual-words method used so far, is that it does not take into account any spatial relationships. The scene with the soldier takes place for the most part in the center of the screen, so to detect this scene spatial information could be useful. A method that uses spatial relationships with bags-of-visual-words features is spatial pyramid matching [33]. This method divides an image into layers. The first layer is the original image, the second layer divides the image into four equal-sized cells and so on for any additional layers. For each cell, a bag-of-visual-words histogram is calculated. Histograms of smaller cells get a higher weight. The concatenation of all histograms forms the feature vector for a sample. A spatial pyramid with one layer is essentially just a regular bags-of-visual-words histogram.

Spatial pyramids with two and three layers were used to train and test SVMs with a linear kernel, a RBF kernel and a histogram intersection kernel. The same videos were used to train and test these SVMs as for the SVMs described earlier in this Subsubsection. The only difference in the training and test data was that the overlay screens were now also filtered out in videos not containing the scene with the soldier. A three-layer pyramid resulted in a decrease in performance for the linear and RBF SVMs and an increase of about 2% for the histogram intersection SVM. A two-layer pyramid resulted in comparable results to using no spatial pyramid matching for the linear and RBF SVMs and an increase of around 4% for the histogram intersection SVM. A two-layer pyramid still looks at a quarter of the frame in each cell in the second layer, so it still considers a large area of the frames and only gives limited information about spatial relationships in frames. The reduced performance of three-layer pyramids suggests that the spatial relationships are either not detected by the method or just not relevant enough to aid in classifying which scene takes place. The best results were achieved using a two-layer spatial pyramid with a histogram intersection SVM, with a recall of 68.3%. A two-layer pyramid achieved better results than a three-layer pyramid, so four-layer pyramids were not tested. Four-layer or even deeper pyramids would also become very computationally expensive, which is another reason these were not tested. The full results are presented in Subsubsection 6.2.2.

A recall of 68.3% was still lower than desired, so the bag-of-visual-words histograms were used as input to train multi-layer perceptrons, a type of neural networks. Unfortunately, these did not lead to an improvement in the results. Various parameters of the network were tried as well as different training algorithms, RPROP [48] and regular back-propagation. Different combinations of the number of hidden layers and the number of hidden nodes per layer were tried, but to no avail. Networks that were trained with a number of samples per class equal to the distribution in the test set, approximately 1 to 10, almost always predicted the negative class. Increasing the ratio to approximately 1 to 4 by decreasing the intervals between processed frames for the positive samples led to similar results. An explanation could be that the nodes "die" during training, where their weights reach 0, from which the network cannot recover. Another explanation could be that the network simply learns the bias to negative samples in the training set and therefore always predicts that the scene does not take place. Unfortunately, most networks trained on an equal number of positive and negative samples in the training set performed poorly as well. Many configurations also predicted predominantly the same class, but even the ones that did not were not suited for determining if the scene takes place. The best network achieved a recall of over 97.9% and a true negative rate of 76.8%. However, because there are more negative test samples than positive, the precision of this network is still only 28.5%, which is just too low to use the method in practice. The full results of the best-performing network are given in Subsubsection 6.2.2. It is possible that bag-of-visual-words histograms are just not good enough features to uniquely detect which scene takes place. The location is the same for all possible scenes, so even after filtering out the UI, many of the detected visual words are still the same for every scene. SVMs also struggled to separate the scenes, so the differences in detected visual words could just be too small to uniquely identify a scene. Additionally, even in the supermarket location, there are multiple unique views that have differing words, making the number of training samples from which unique visual words can be distinguished even smaller. Bags-of-visual-words histograms did work well to identify the supermarket location, but the differences between the supermarket and other locations in the game are larger, so there it is easier to find descriptive visual words.

To get satisfactory results an even more complex method was needed. For this purpose convolutional neural networks were used. They were trained in Python using the Microsoft Cognitive Toolkit (CNTK) [3]. All networks discussed here were trained via transfer learning from a CNTK implementation of AlexNet[1] [1], which is based on work by Krizhevsky et al. [30]. Initially networks were trained that intended to solve both the second and third research questions at once with various parameters to find a good configuration. However these netwerks did not perform that well and the choice to kill both the soldier and the woman was never predicted correctly on the testset. Therefore the choice to kill both the soldier and the woman was no longer considered separately. More information about using convolutional neural networks to detect the player's choice can be found in Subsection 5.4. To detect whether a supermarket visit contains a scene with the soldier a convolutional neural network was trained via transfer learning on frames

[1] https://www.cntk.ai/Models/CNTK_Pretrained/AlexNet_ImageNet_CNTK.model

from supermarket visits not containing the scene with the soldier and frames that do contain it. All frames were resized to 227 by 227 RGB images, as that was the required input size for the network used for transfer learning. The training set consisted of every 30th frame from 80 videos without the scene, every 3rd frame from 25 videos where the player intervenes and every 2nd frame from 10 videos where the player remains passive. Samples from videos where the scene takes place only include frames that actually occur during the scene, not from the entire supermarket visit. The samples from videos where the player intervenes or remains passive together form the training samples for videos containing the scene with the soldier. The smaller intervals in these videos were chosen to have similar numbers of samples per class in an unbalanced dataset, which contains more videos without the scene with the soldier than with it. Having a similar number of samples per class is desirable to reduce the risk of overfitting to the dominant class. Reducing the frame interval between frames used for training in videos containing the scene is essentially a variation of oversampling. The benefit over real oversampling is that all samples are still from real videos with small differences between them. Real oversampling would add synthetic samples by duplicating existing samples or adding permutations of images, for example by rotating them. The same intervals were used for the test set, which consisted of 40 videos without the scene and 18 videos with the scene, divided into 13 videos where the player intervenes and 5 where the player remains passive. A new network was trained using transfer learning on the CNTK implementation of AlexNet[2] [1] referred to earlier. The network was trained for 30 epochs with a learning rate of 0.002 for the first ten epochs and 0.001 afterwards. It correctly predicted all training samples and achieved a recall of approximately 78.2% on the test set. The full results are presented in Subsubsection 6.2.2.

5.3.2 Determining if a supermarket visit contains a scene with the soldier

In the previous Subsubsection a convolutional neural network was discussed that achieved the best results of all tested methods on classifying whether frames contain the scene with the soldier. By itself this is not enough to determine whether a supermarket visit contains the scene with the soldier. However, when classifications of the convolutional neural network are combined with some additional rules, scenes with the soldier can be detected reasonably reliably. To do so, a number of frames per supermarket visit must be processed and the network must predict their class. Then per video the percentage of frames, which are classified as displaying the scene, is calculated. If this percentage exceeds an experimentally set threshold, the video is judged to contain the scene with the soldier. This method was tested on 58 videos, 40 of which did not contain the scene with the soldier and 18 that did. Every 30th frame of these videos was processed, but the overlay screens were filtered out. The threshold was set at 0.2. With these settings, for 57 out of the 58 videos in the test set, the correct class could be determined. The full results are presented in Subsubsection 6.2.3

5.4 Detecting the player's choice

Analogous to detecting whether the scene with the soldier takes place, determining what choice the player makes in such a scene can be split into two sub-problems. The first is to classify what choice individual frames depict. This is discussed in Subsubsection 5.4.1. The second sub-problem is to determine which choice the player makes in a scene with

[2] https://www.cntk.ai/Models/CNTK_Pretrained/AlexNet_ImageNet_CNTK.model

the soldier. This corresponds to the third research question and a solution is discussed in Subsubsection 5.4.2.

5.4.1 Determining what choice a frame depicts

To determine what choice a frame depicts the same method was used as to detect whether the scene takes place: using convolutional neural networks trained via transfer learning on a Microsoft Cognitive Toolkit (CNTK) [3] implementation of AlexNet[3] [1], based on the original by Krizhevsky et al. [30]. The method described in the previous Subsection is used in an iterative fashion: first the scene is detected and then using the same approach the choice of the player can be detected. As was mentioned in Subsubsection 5.3.1, none of the trained networks could correctly predict any of the test samples representing the choice to kill both the soldier and the woman. Therefore, detecting what choice the player makes has been reduced to a binary problem, where the only possible choices are to intervene or to remain passive. Initially a network was trained where frames from the entire scene were used for both of the classes. However, this caused confusion between the classes, because the two scenes are practically the same until the point where the actual intervention takes place. Therefore a new network was trained where the training samples for the choice to intervene consisted only of frames depicting the actual intervention. This was defined as opening the door the the room with the soldier and the woman and then moving towards them. In most cases this was followed by the player attacking the soldier, although it is also possible to run away, at which point the woman also runs away. The training set consisted of 25 videos where the player intervenes. Every frame where the intervention was in progress was processed here. It also contained 10 videos where the player remains passive. Here, every 2nd frame was processed from the entire scene with the soldier. The same intervals were used for the test set, which contained 13 videos where the player intervened and 5 where the player remained passive. The network was trained for 30 epochs with a learning rate of 0.002 for the first ten epochs and 0.001 afterwards, the same parameters as for detecting the scene. 100% of the frames were correctly predicted on the training set and on the test set approximately 84.1% of the frames were correctly predicted. The main thing that stands out is that while the recall is high at 98.2%, the precision is only 69.2%. The most likely cause for this is that the frames that depict an intervention show many similarities to frames where the player remains passive. The location is the same in both cases and they both contain the soldier and the woman. The only thing that differs are the actions of the player. These only take up a minority of the screen, which means they may not have enough effect to distinguish the frames where the player remains passive from those where the player intervenes. A tailor-made convolutional neural network may be able to achieve better results, but would also take much more time to set up. Another reason could be that there are not that many instances where the player remains passive. Although, there are enough frames, these are often similar because the intervals are small. If there were more unique videos available with the scene with the soldier to train on, the variation in training samples for each class would increase, which may make the two classes easier to separate. The full results are presented in Subsubsection 6.3.2.

5.4.2 Determining what choice the player makes in a scene with the soldier

The convolutional neural network described in the previous section can be used to detect which choice the player makes during a scene with the soldier. To do so, a method can be

[3] https://www.cntk.ai/Models/CNTK_Pretrained/AlexNet_ImageNet_CNTK.model

used similar to the one described in Subsubsection 5.3.2, which was used to determine if a supermarket visit contains a scene with the soldier. Per video, a percentage is calculated, that represents how many frames are classified as displaying an intervention. If this percentage exceeds an experimentally set threshold of 0.09, the video is judged to display a player intervening during the scene with the soldier. This method was tested on 18 videos, 13 of which contained an intervention and 5 where the player remained passive. Every 30th frame was processed during scenes with the soldier. Only frames during the scene were processed, not other frames during the same supermarket visit. For 17 out of 18 videos in the test set, the correct class can be determined using this method. The full results are presented in Subsubsection 6.3.3.

6 Experiments

The experiments that were performed to test the developed methods described in Section 5 are discussed in this section along with their results. The experiments regarding the detection of the supermarket are discussed in Subsection 6.1. Experiments about detecting the scene with the soldier are discussed in Subsection 6.2. Lastly, the experiments regarding the choice of the player are discussed in Subsection 6.3.

The following abbreviations are used in the tables in this section, where TP stands for the number of true positives, FP for the number of false positives, TN for the number of true negatives and FN for the number of false negatives:

- Rec = recall/true positive rate = $\frac{TP}{TP+FN}$
- FPR = false positive rate = $\frac{FP}{FP+TN}$
- TNR = true negative rate = $\frac{TN}{TN+FP}$
- FNR = false negative rate = $\frac{FN}{FN+TP}$
- Prec = precision = $\frac{TP}{TP+FP}$
- Acc = accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$
- F1 = F1-score = $\frac{2*TP}{2*TP+FP+FN}$

6.1 Detecting the supermarket

Detecting the supermarket was split into two separate sub-problems: determining whether a frame displays the supermarket and determining when a visit to the supermarket begins and ends in a video. Several datasets were used to test the methods developed to solve these problems. These datasets are described in Subsubsection 6.1.1. The results regarding classifying whether a frame contains the supermarket are presented in Subsubsection 6.1.2 and regarding determining the beginning and ending of a visit to the supermarket in Subsubsection 6.1.3. In this Subsection for the purposes of calculating statistics such as precision and recall, segments with the supermarket were considered as positive samples and videos without it as negative samples.

6.1.1 Description of datasets

To predict whether a frame contains the supermarket, both a SVM and a K-nearest neighbor classifier were trained using features consisting of bag-of-visual-words histograms of SIFT descriptors as described in Subsubsection 5.2.1. Several datasets were used to test both these classifiers. Initially only the training and main test sets were used. Later the extended sets and additional test sets were added to test the performance of the program on cases more realistic to handling unseen data after this project has been completed. The extended training and test sets were also used to test the developed methods that determine the beginning and ending of a visit to the supermarket. All datasets used in this Subsection are described below with their abbreviations used in the tables in parentheses:

- Training set (Train): the training set consisted of 50 segments that contain a visit to the supermarket as its positive samples. Features were computed for every 30th frame. Frames not containing the supermarket or that contain an overlay screen or screen fading were excluded. 30 videos not containing the supermarket were used as the negative samples. Here, features were computed for every 60th frame. There were 11435 positive samples and 12341 negative samples.

- Main test set (Test 1): the main test set consisted of 50 segments that contain a visit to the supermarket as its positive samples. Features were computed for every 30th frame. Frames not containing the supermarket or that contain an overlay screen or screen fading were excluded. 20 videos not containing the supermarket were used as the negative samples. Here, features were computed for every 30th frame. There were 12028 positive samples and 19901 negative samples.

- Second test set (Test 2a): the second test set consisted of 12 previously unseen videos where the supermarket was visited once. Features were computed for every 30th frame, including frames that did not display the supermarket or frames that contained an overlay screen. Features were not computed for frames occurring during screen fading as these do not have a clear label[4]. There were 2248 positive samples and 14696 negative samples.

- Second test set overlay screens (Test 2b): the second test set, but now only the frames displaying overlay screens. There were 859 negative samples. This test set and the test set described below (second test set without supermarket) were used to determine whether the false positives were primarily caused by the overlay screens, by frames not containing the supermarket or more or less evenly distributed across both.

- Second test set without supermarket (Test 2c): the second test set, but now only the frames occurring before and after a visit to the supermarket. There were 13837 negative samples.

- Third test set(Test 3): the remaining videos that were not yet used in the other test or training sets. All these videos contain multiple visits to the supermarket for a total of 13 visits to the supermarket. Features were computed for every 30th frame containing the supermarket for a total of 1834 positive samples.

- Second test set frame averages (Test 2 Avg): a test set used to test if calculating features for the average of all 30 frames in an interval instead of only calculating

[4] In these frames the supermarket fades in or out. Here it is not clear at what degree of fading the supermarket is clearly visible and what the frame's label should be.

features for every 30th frame would give better results. The same data was used as for the regular second test set. Because all 30 frames were needed to calculate an average there are slightly fewer samples than in the regular second test set. There were 2246 positive samples and 14686 negative samples.

- Negative test set frame averages (Test Neg Avg): a test set used to test if calculating features for the average of all 30 frames in an interval instead of only calculating features for every 30th frame would give better results. The test set consisted of 10 videos not containing the supermarket that were also used in the main test set. The averages were calculated of every 30 frames. There were 11178 negative samples.

- Unrelated videos (Unrelated): 14 videos not containing footage of 'This War of Mine' where features were computed for every 200th frame. These videos violate the assumption made in Subsection 4.1 that all videos being processed contain footage of 'This War of Mine'. This dataset was used to test how well the method performs if this assumption is violated or how well it generalizes without performing any adaptation to other domains. There were 1758 negative samples.

- Extended training set(Train Ext): the same videos as in the regular training set, but now every 30th frame of the positive samples was used, except for frames occurring during screen fading as these do not have a clear label (i.e. frames before and after the appearance and disappearance of the supermarket were also processed in these videos, instead of only frames displaying the supermarket). Additionally, every 30th frame of the negative videos was used. The frames that were added here were unseen during the training of SVMs on the regular training set, so in the tables below for the SVM with a linear kernel this dataset consists of a mixture of samples seen during training and unseen negative samples. There were 11435 positive samples and 67277 negative samples.

- Extended main test set (Test Ext): the same videos as in the regular test set, but now every 30th frame of every video was used, except for frames occurring during screen fading in positive samples as these do not have a clear label. There were 12028 positive samples and 63029 negative samples.

- Scene times training set: the extended training set, but now frames occurring during screen fading are included as well. This mimics an unseen video being processed after completion of this project. This dataset was used to determine the thresholds for the method described in Subsubsection 5.2.2.

- Scene times test set: the extended main test set, but now frames occurring during screen fading are included as well for the same reason as in the scene times training set. This dataset was used to test the method described in Subsubsection 5.2.2.

6.1.2 Classifying frames

The results of the SVM classifier with a linear kernel using bag-of-visual-words histograms of SIFT descriptors on all test sets are presented in Table 1. These same features were also used to train a K-nearest neighbor classifier, of which the results are given in Table 2. As the results of the K-nearest neighbor classifier were considerably lower than of the SVM, the SVM was selected for further processing. This is also the reason the K-nearest neighbor classifier has not been tested on all datasets (as several of these were not developed until after the choice to use the SVM for further processing was made).

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train	0.99659	0.00583	0.99417	0.00341	0.99372	0.99533	0.99515
Test 1	0.94505	0.03794	0.96206	0.05496	0.93772	0.95565	0.94137
Test 2a	0.83185	0.03048	0.96952	0.16815	0.80673	0.95125	0.81910
Test 2b	N/A	0.01397	0.98603	N/A	N/A	0.98603	N/A
Test 2c	N/A	0.03158	0.96842	N/A	N/A	0.96842	N/A
Test 3	0.94820	N/A	N/A	0.05180	N/A	0.94820	0.97341
Test 2 Avg	0.77605	0.03915	0.96085	0.22395	0.75194	0.93633	0.76380
Test Neg Avg	N/A	0.06800	0.93200	N/A	N/A	0.93200	N/A
Unrelated	N/A	0.19056	0.80944	N/A	N/A	0.80944	N/A
Train Ext	0.99659	0.03548	0.96542	0.00341	0.82682	0.96918	0.90380
Test Ext	0.94505	0.03925	0.96075	0.05496	0.82126	0.95823	0.87881

Table 1: Results of the SVM with a linear kernel on all datasets before 'temporal smoothing'. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train	0.83542	0.01629	0.98371	0.16458	0.97939	0.91239	0.90169
Test 1	0.69081	0.01629	0.95709	0.30920	0.90680	0.85678	0.78420
Test 2a	0.67082	0.02443	0.97557	0.32918	0.80771	0.93514	0.73293
Test 3	0.70229	N/A	N/A	0.29771	N/A	0.70229	0.82511
Unrelated	N/A	0.03641	0.96360	N/A	N/A	0.96360	N/A

Table 2: K-nearest neighbor classifier results on 5 datasets. K was set to the square root of the number of training samples. All results are rounded to 5 decimals.

Table 1 also contains the results of the SVM with a linear kernel on features calculated for the averages of frame intervals instead of only every 30th frame. As can be seen in the table, this method turned out to perform worse. Therefore, 'temporal smoothing' was developed to improve the classification results. The results of this method on all datasets where no frame averaging had been performed are presented in Tables 3 up to and including Table 11. From these results it can be seen that a window size of 2 achieved the best results on the majority of the datasets, which is why it was chosen as the final window size.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	0.99659	0.00583	0.99417	0.00341	0.99372	0.99533	0.99515
Window 10	0.99756	0.00332	0.99668	0.00245	0.99642	0.99710	0.99699
Window 8	0.99808	0.00332	0.99678	0.00192	0.99642	0.99735	0.99725
Window 6	0.99825	0.00324	0.99676	0.00175	0.99651	0.99745	0.99738
Window 4	0.99843	0.00308	0.99692	0.00157	0.99668	0.99765	0.99755
Window 2	0.99878	0.00230	0.99700	0.00122	0.99677	0.99786	0.99777

Table 3: Results of the SVM with a linear kernel on the training set after 'temporal smoothing' with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	0.94505	0.03794	0.96206	0.05496	0.93772	0.95565	0.94137
Window 10	0.94837	0.03417	0.96583	0.05163	0.94374	0.95925	0.94605
Window 8	0.94987	0.03337	0.96664	0.05013	0.94507	0.96032	0.94746
Window 6	0.95128	0.03301	0.96699	.04872	0.94570	0.96107	0.94848
Window 4	0.95336	0.03216	0.96784	0.04664	0.94714	0.96239	0.95024
Window 2	0.95627	0.03151	0.96849	0.04373	0.94831	0.96389	0.95227

Table 4: Results of the SVM with a linear kernel on the main test set after ‘temporal smoothing’ with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	0.83185	0.03048	0.96952	0.16815	0.80673	0.95125	0.81910
Window 10	0.83585	0.02885	0.97115	0.16415	0.81589	0.95320	0.82575
Window 8	0.83719	0.02831	0.97169	0.16281	0.81897	0.95385	0.82798
Window 6	0.83630	0.02735	0.97265	0.16370	0.82384	0.95456	0.83002
Window 4	0.83541	0.02749	0.97251	0.16459	0.82296	0.95432	0.82914
Window 2	0.83585	0.02735	0.97265	0.16415	0.82376	0.95450	0.82976

Table 5: Results of the SVM with a linear kernel on the second test set after ‘temporal smoothing’ with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	N/A	0.01397	0.98603	N/A	N/A	0.98603	N/A
Window 10	N/A	0.01397	0.98603	N/A	N/A	0.98603	N/A
Window 8	N/A	0.01281	0.98719	N/A	N/A	0.98719	N/A
Window 6	N/A	0.01281	0.98719	N/A	N/A	0.98719	N/A
Window 4	N/A	0.01397	0.98603	N/A	N/A	0.98603	N/A
Window 2	N/A	0.01513	0.98487	N/A	N/A	0.98487	N/A

Table 6: Results of the SVM with a linear kernel on only the overlay screens of the second test set after ‘temporal smoothing’ with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	N/A	0.03158	0.96842	N/A	N/A	0.96842	N/A
Window 10	N/A	0.02941	0.97059	N/A	N/A	0.97059	N/A
Window 8	N/A	0.02884	0.97116	N/A	N/A	0.97116	N/A
Window 6	N/A	0.02782	0.97218	N/A	N/A	0.97218	N/A
Window 4	N/A	0.02761	0.97239	N/A	N/A	0.97239	N/A
Window 2	N/A	0.02732	0.97268	N/A	N/A	0.97268	N/A

Table 7: Results of the SVM with a linear kernel on only the frames before the start and after the end of a visit to the supermarket in the second test set after ‘temporal smoothing’ with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	0.94820	N/A	N/A	0.05180	N/A	0.94820	0.97341
Window 10	0.94984	N/A	N/A	0.05016	N/A	0.94984	0.97427
Window 8	0.95256	N/A	N/A	0.04744	N/A	0.95256	0.97571
Window 6	0.95420	N/A	N/A	0.04580	N/A	0.95420	0.97656
Window 4	0.95638	N/A	N/A	0.04362	N/A	0.95638	0.97770
Window 2	0.95638	N/A	N/A	0.04362	N/A	0.95638	0.97770

Table 8: Results of the SVM with a linear kernel on the third test set after 'temporal smoothing' with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	N/A	0.19056	0.80944	N/A	N/A	0.80944	N/A
Window 10	N/A	0.17463	0.82537	N/A	N/A	0.82537	N/A
Window 8	N/A	0.17463	0.82537	N/A	N/A	0.82537	N/A
Window 6	N/A	0.17065	0.82935	N/A	N/A	0.82935	N/A
Window 4	N/A	0.16212	0.83788	N/A	N/A	0.83788	N/A
Window 2	N/A	0.15472	0.84528	N/A	N/A	0.84528	N/A

Table 9: Results of the SVM with a linear kernel on the unrelated videos test set after 'temporal smoothing' with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	0.99659	0.03548	0.96452	0.00341	0.82682	0.96918	0.90380
Window 10	0.99755	0.03373	0.96627	0.00245	0.83409	0.97082	0.90853
Window 8	0.99799	0.03356	0.96644	0.00201	0.83482	0.97102	0.90914
Window 6	0.99808	0.03294	0.96706	0.00192	0.83741	0.97157	0.91071
Window 4	0.99825	0.03277	0.96723	0.00175	0.83811	0.97173	0.91120
Window 2	0.99834	0.03190	0.96810	0.00166	0.84176	0.97250	0.91339

Table 10: Results of the SVM with a linear kernel on the extended training set after 'temporal smoothing' with different window sizes. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
No smoothing	0.94505	0.03925	0.96075	0.05496	0.82126	0.95823	0.87881
Window 10	0.94812	0.03608	0.96392	0.05188	0.83375	0.96139	0.88726
Window 8	0.94962	0.03549	0.96451	0.05038	0.83623	0.96212	0.88932
Window 6	0.95062	0.03489	0.96511	0.04938	0.83870	0.96279	0.89116
Window 4	0.95195	0.03417	0.96583	0.04805	0.84166	0.96360	0.89341
Window 2	0.95261	0.03348	0.96652	0.04739	0.84449	0.96429	0.89530

Table 11: Results of the SVM with a linear kernel on the extended main test set after 'temporal smoothing' with different window sizes. All results are rounded to 5 decimals.

The SVM with a linear kernel performed well on its own, but the false positives were often clustered, which led to the method described in Subsubsection 5.2.2 erroneously detecting scenes from these clustered false positives. A SVM with a Radial Basis Function trained on the extended training set classified fewer frames without the supermarket as containing the supermarket. The results of this SVM can be found in Table 12.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train	0.99650	0.00016	0.99984	0.00350	0.99983	0.99823	0.99816
Test 1	0.87313	0.01050	0.98950	0.12687	0.98045	0.94566	0.92370
Test 2a	0.73754	0.00871	0.99129	0.26246	0.92833	0.95763	0.82201
Test 2b	N/A	0	1	N/A	N/A	1	N/A
Test 2c	N/A	0.00867	0.99133	N/A	N/A	0.99133	N/A
Test 3	0.94875	N/A	N/A	0.05125	N/A	0.94875	0.97370
Unrelated	N/A	0.04039	0.95961	N/A	N/A	0.95961	N/A
Train Ext	0.99502	0.00245	0.99755	0.00498	0.98571	0.99718	0.99034
Test Ext	0.86448	0.01036	0.98964	0.13552	0.94091	0.96958	0.90108

Table 12: Results of the SVM with a Radial Basis Function (RBF) kernel on various datasets after ‘temporal smoothing’ with window size 2. All results are rounded to 5 decimals.

6.1.3 Start and end frames of a supermarket visit

The method described in Subsubsection 5.2.2 was used to detect the start and end frames of supermarket visits based on the classified frames from two SVMs, one with a linear kernel and one with a Radial Basis Function (RBF) kernel. The results of the used method on the scene times training set can be found in Table 13. It contains the results for the frames classified by both of the SVMs. A scene start or end was considered to be correctly identified when it fell within 90 frames of the actual start of the annotations. The 90 frame margin was chosen for two reasons. First, screen fading can occur which can lead to incorrect predictions. Second, the annotations consisted of the start and end times in seconds. These times were then converted to frames using a function from OpenCV’s VideoCapture class. However, because the annotations were in seconds and there are generally 30 frames per second in the videos in the dataset, the annotations can be slightly off the true start or end frames of a supermarket visit. When a scene start has been identified too early or too late, four categories were used to indicate how large the discrepancy between the predicted and true start or end was. Only if at least 1 scene start or end fell in a category, was the category incorporated in the table below. The table also includes entries for the overall numbers of scene starts and ends that were predicted too early or too late.

	SVM with linear kernel	SVM with RBF kernel
Videos without supermarket where no scene was detected	27	30
Scene starts correctly identified	48	48
Scene starts identified less than 500 frames too early	1	1
Scene starts identified too early overall	1	1
Scene starts identified between 1000 and 2000 frames too late	1	1
Scene starts identified too late overall	1	1
Scene starts identified where no scene occurred	21	0
Scene ends correctly identified	42	49
Scene ends identified less than 500 frames too early	2	1
Scene ends identified too early overall	2	1
Scene ends identified less than 500 frames too late	2	0
Scene ends identified between 500 and 1000 frames too late	2	0
Scene ends identified between 1000 and 2000 frames too late	1	0
Scene ends identified more than 2000 frames too late	1	0
Scene ends identified too late overall	6	0
Scene ends identified where no scene occurred	21	0

Table 13: Results of the supermarket visit detection method on the videos in the scene times training set after classification by a SVM with a linear kernel and a SVM with a Radial Basis Function (RBF) kernel. There were 50 supermarket visits and 30 videos without the supermarket in the dataset used here.

In Table 13 it can be seen that the frames classified by the SVM with the RBF kernel achieve much better results in identifying scene starts and ends on the training set. The results for the scene times test set are presented in Table 14. In this table an additional category has been added for scenes that have been detected as two separate scenes instead of one. The videos in this category are also still incorporated into scene starts and ends that were predicted too early or too late, but as these are not false positives, they are not incorporated into scene starts or ends that were identified when no scene occurred. The SVM with the RBF kernel also performs better on the test set, especially in not detecting as many scenes incorrectly.

	SVM with linear kernel	SVM with RBF kernel
Videos without supermarket where no scene was detected	14	17
Scene starts correctly identified	49	47
Scene starts identified between 1000 and 2000 frames too early	0	1
Scene starts identified too early overall	0	1
Scene starts identified less than 500 frames too late	0	1
Scene starts identified between 500 and 1000 frames too late	1	0
Scene starts identified more than 2000 frames too late	0	1
Scene starts identified too late overall	1	2
Scene starts identified where no scene occurred	23	8
Scene starts identified that were part of a previous scene	2	3
Scene ends correctly identified	40	39
Scene ends identified less than 500 frames too early	0	1
Scene ends identified between 500 and 1000 frames too early	0	1
Scene ends identified between 1000 and 2000 frames too early	1	1
Scene ends identified more than 2000 frames too early	4	6
Scene ends identified too early overall	5	9
Scene ends identified less than 500 frames too late	1	0
Scene ends identified between 500 and 1000 frames too late	2	2
Scene ends identified between 1000 and 2000 frames too late	1	0
Scene ends identified more than 2000 frames too late	1	0
Scene ends identified too late overall	5	2
Scene ends identified where no scene occurred	23	8
Scene ends identified that were part of a previous scene	2	3

Table 14: Results of the supermarket visit detection method on the videos in the scene times test set after classification by a SVM with a linear kernel and a SVM with a Radial Basis Function (RBF) kernel. There were 50 supermarket visits and 20 videos without the supermarket in the dataset used here.

6.2 Detecting the scene with the soldier

Detecting whether the scene with the soldier takes place was split into two sub-problems: determining whether a frame displays the scene and determining whether a supermarket visit contains a scene with the soldier. The datasets used to train and test the methods used to solve these problems are described in Subsubsection 6.2.1. The results regarding classifying whether a frame does or does not display the scene are presented in Subsubsection 6.2.2. The results of determining whether a supermarket visit contains a scene with the soldier are given in Subsubsection 6.2.3. In this Subsection for the purposes of calculating statistics such as precision and recall, segments with the scene were considered as positive samples and segments without it as negative samples.

6.2.1 Description of datasets

Various datasets were used to train and test SVMs, multi-layer perceptrons and convolutional neural networks to detect the scene with the soldier. These datasets are described below:

- SVM training set (Train): the training set used for SVMs with bag-of-visual-words features consisted of 18 segments where the scene with the soldier occurred and 40 where it did not. In all segments every 30th frame was processed. Frames not containing the supermarket or an overlay screen were filtered out in the segments containing the scene. There were 1217 samples with the scene and 14735 without it.

- SVM test set (Test): the test set used for SVMs with bag-of-visual-words features consisted of 19 segments where the scene with the soldier occurred and 40 where it did not. In all segments every 30th frame was processed. Frames not containing the supermarket or an overlay screen were filtered out in the segments containing the scene. There were 1062 samples with the scene and 11268 without it.

- SVM region-of-interest training set (Train ROI): the training set used for SVMs with bag-of-visual-words features from frames where the user interface had been filtered out consisted of 18 segments where the scene with the soldier occurred and 40 where it did not. In all segments every 30th frame was processed. Frames not containing the supermarket or an overlay screen were filtered out in the segments containing the scene. There were 1217 samples with the scene and 14735 without it.

- SVM/MLP region-of-interest test set (Test ROI): the test set used for SVMs and multi-layer perceptrons with bag-of-visual-words features from frames where the user interface had been filtered out consisted of 19 segments where the scene with the soldier occurred and 40 where it did not. In all segments every 30th frame was processed. Frames not containing the supermarket or an overlay screen were filtered out in the segments containing the scene. There were 1062 samples with the scene and 11268 without it.

- SPM SVM training set (Train SPM): the training set used for SVMs with spatial pyramid matching (SPM). Bag-of-visual-words features were extracted from frames where the user interface had been filtered out. It consisted of 18 segments where the scene with the soldier occurred and 40 where it did not. In all segments every 30th frame was processed. Frames not containing the supermarket or an overlay screen were filtered out in all segments. There were 1217 samples with the scene and 9857 without it.

- SPM SVM test set (Test SPM): the training set used for SVMs with spatial pyramid matching (SPM). Bag-of-visual-words features were extracted from frames where the user interface had been filtered out. It consisted of 19 segments where the scene with the soldier occurred and 40 where it did not. In all segments every 30th frame was processed. Frames not containing the supermarket or an overlay screen were filtered out in all segments. There were 1062 samples with the scene and 7486 without it.

- MLP training set equal (Train MLP equal): the training set used to train multi-layer perceptrons, where the number of samples for both classes has been made equal. The same data was used as for the SVM training sets described above. For the samples without the scene a random subset was taken from all samples in order to have an equal number of samples per class. There were 1217 samples with the scene and 1217 without it.

- CNN training set (Train CNN): the training set used to train a convolutional neural network using transfer learning. It consisted of every 30th frame from 80 segments without the scene and 35 segments with the scene. These segments contain every 3rd frame from 25 segments where the player intervened and every 2nd frame from 10 segments where the player remained passive. All frames were resized to 227 by 227 pixels. The overlay screens were filtered out in all segments. There were 26828 samples with the scene and 18838 samples without it.

- CNN test set (Test CNN): the test set used to evaluate a convolutional neural network that was trained using transfer learning. It consisted of every 30th frame from 40 segments without the scene and 18 segments with the scene. These segments contain every 3rd frame from 13 segments where the player intervened and every 2nd frame from 5 segments where the player remained passive. All frames were resized to 227 by 227 pixels. The overlay screens were filtered out in all segments. There were 12868 samples with the scene and 7486 samples without it.

- Scene detection test set (Test Scene): the test set used to test the method that detects whether the scene with the soldier occurs based on the convolutional neural network predictions. It consisted of the same 58 segments as in the CNN test set, but here for every segment every 30th frame was processed for the entire segment. I.e. for segments containing the scene frames before and after the scene were also processed. The overlay screens were filtered out in all segments. There were 18 segments with the scene and 40 without it.

6.2.2 Classifying frames

The results of the SVM classifier with a linear kernel and a RBF kernel using bag-of-visual-words histograms of SIFT descriptors are presented in Table 15. As both of these SVMs performed poorly and many detected keypoints and descriptors were part of the user interface (UI), additional SVMs were trained on data where the UI had been filtered out. The bag-of-visual-words codebooks used to generate features for these SVMs had also been clustered on data with the UI filtered out. This filtering was done using a region-of-interest (ROI) image mask. SVMs trained on data where the UI had been filtered out were trained with three different kernels: a linear kernel, a RBF kernel and a histogram intersection kernel. Their results are given in Table 16.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train linear	0.99589	0.00081	0.99919	0.00411	0.99020	0.99893	0.99304
Test linear	0.42467	0.06425	0.93575	0.57533	0.38383	0.89173	0.40322
Train RBF	0.84717	0.00265	0.99735	0.15284	0.96355	0.98590	0.90162
Test RBF	0.38418	0.05591	0.94409	0.61582	0.39306	0.89586	0.38857

Table 15: Results of the SVMs with a linear kernel and a RBF kernel on the training and test sets for detecting whether frames display the scene with the soldier. All results are rounded to 5 decimals.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train ROI linear	0.98603	0	1	0.01397	1	0.99893	0.99297
Test ROI linear	0.62241	0.01917	0.98083	0.37759	0.75371	0.94996	0.68180
Train ROI RBF	0.98603	0	1	0.01397	1	0.99893	0.99297
Test ROI RBF	0.61111	0.01677	0.98323	0.38889	0.77446	0.95118	0.68316
Train ROI inter	1	0	1	0	1	1	1
Test ROI inter	0.64878	0.01100	0.98900	0.35122	0.84748	0.95969	0.73493

Table 16: Results of the SVMs with a linear kernel, a RBF kernel and a histogram intersection kernel (inter) on the training and test sets for detecting whether frames display the scene with the soldier, where the UI has been filtered out of the data. All results are rounded to 5 decimals.

To try to improve the results spatial pyramid matching was used with the bag-of-visual-words features extracted from frames where the UI had been filtered out. The codebooks used to generate these histograms were the same as those used to generate histograms for the regular region-of-interest bag-of-visual-words histograms (whose results were presented in Table 16). Spatial pyramids consisting of two and three layers were used to train SVMs. These SVMs were trained using a linear, a RBF and a histogram intersection kernel. Their results are presented in Table 17.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train SPM 2-layer linear	0.97453	0	1	0.02547	1	0.99720	0.98710
Test SPM 2-layer linear	0.61864	0.03086	0.96914	0.38136	0.73987	0.92560	0.67385
Train SPM 3-layer linear	1	0	1	0	1	1	1
Test SPM 3-layer linear	0.50283	0.04168	0.95832	0.49718	0.63121	0.90173	0.55975
Train SPM 2-layer RBF	0.97206	0	1	0.02794	1	0.99693	0.98583
Test SPM 2-layer RBF	0.62429	0.02471	0.97529	0.37571	0.78184	0.93168	0.69424
Train SPM 3-layer RBF	1	0	1	0	1	1	1
Test SPM 3-layer RBF	0.43597	0.02044	0.97956	0.56403	0.75162	0.91203	0.55185
Train SPM 2-layer inter	1	0	1	0	1	1	1
Test SPM 2-layer inter	0.68267	0.00975	0.99025	0.31733	0.90852	0.95204	0.77957
Train SPM 3-layer inter	1	0	1	0	1	1	1
Test SPM 3-layer inter	0.66855	0.00962	0.99038	0.33145	0.90793	0.95040	0.77007

Table 17: Results of the SVMs with a linear kernel, a RBF kernel and a histogram intersection kernel (inter) on the training and test sets for detecting whether frames display the scene with the soldier, where the UI has been filtered out of the data and spatial pyramid matching (SPM) was used. All results are rounded to 5 decimals.

The results of SVMs with spatial pyramid matching were still not satisfactory. Therefore, neural networks were tried instead. At first, many forms of multi-layer perceptrons were trained and tested, but none of these performed well. The results of the best-performing multi-layer perceptron are presented in Table 18. As these proved to not yield results of sufficient quality, convolutional neural networks were used instead. The confusion matrices of the convolutional neural network described in Subsubsection 5.3.1 for the training and test sets are presented in Tables 19 and 20. The statistics of this network are presented in Table 21.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train MLP equal	0.99754	0.00493	0.99507	0.00247	0.99508	0.99630	0.99631
Test ROI	0.97928	0.23181	0.76819	0.02072	0.28478	0.78638	0.44124

Table 18: Results of the best-performing multi-layer perceptron on the training and test sets for detecting whether frames display the scene with the soldier. All results are rounded to 5 decimals.

		Predicted class		
		Scene	**No scene**	**Total**
True class	**Scene**	26828	0	26828
	No scene	0	18838	18838
	Total	26828	18838	

Table 19: Confusion matrix of the predictions of a convolutional neural network on the CNN training set.

		Predicted class		
		Scene	**No scene**	**Total**
True class	**Scene**	10059	2809	12868
	No scene	150	7336	7486
	Total	10209	10145	

Table 20: Confusion matrix of the predictions of a convolutional neural network on the CNN test set.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train CNN	1	0	1	0	1	1	1
Test CNN	0.78171	0.02004	0.97996	0.21829	0.98531	0.85462	0.87178

Table 21: Results of the convolutional neural network trained via transfer learning on the training and test set for detecting whether the scene with the soldier takes place. All results are rounded to 5 decimals.

6.2.3 Scene detection

In Subsubsection 5.3.2 the method was described that was used to predict whether a video contains the scene with the soldier based on the predictions of a convolutional neural network. The results of this method are presented as a confusion matrix in Table 22. The statistics of the method are presented in Table 23.,

		Predicted class		
		Scene	**No scene**	**Total**
True class	**Scene**	18	0	18
	No scene	1	39	40
	Total	19	39	

Table 22: Confusion matrix of the predictions of the scene detection method based on convolutional neural network predictions on the scene detection test set.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Test Scene	1	0.025	0.975	0	0.94739	0.98276	0.97297

Table 23: Results of the scene detection method on the scene detection test set. All results are rounded to 5 decimals.

6.3 Detecting the player's choice

Analogous to detecting the supermarket and the scene with the soldier, detecting the player's choice during a scene with the soldier was split into two sub-problems: determining which choice a frame depicts and and determining which choice is made during a scene with the soldier. The results of the former are presented in Subsubsection 6.3.2 and of the latter in Subsubsection 6.3.3. The datasets used for the experiments are described in Subsubsection 6.3.1. In this Subsection, for the purposes of calculating statistics such as recall and precision, samples from segments where the player intervened were considered positive samples and samples from segments where the player remained passive were considered negative samples.

6.3.1 Description of datasets

The datasets that were used to train and test the methods to detect what choice the player makes are described below:

- First CNN training set (Train CNN): the first training set used to train a convolutional neural network via transfer learning. For all samples the entire scene with the soldier was taken into account. Every 3rd frame was processed in segments where the player intervened and every 2nd frame in segments where the player remained passive. Overlay screens were filtered out in all segments. There were 13458 samples from segments where the player intervened and 13370 samples from segments where the player remained passive.

- First CNN test set (Test CNN): the first test set used to test a convolutional neural network trained via transfer learning. For all samples the entire scene with the soldier was taken into account. Every 3rd frame was processed in segments where the player intervened and every 2nd frame in segments where the player remained passive. Overlay screens were filtered out in all segments. There were 6541 samples from segments where the player intervened and 6327 samples from segments where the player remained passive.

- Second CNN training set (Train CNN 2): the second training set used to train a convolutional neural network via transfer learning. In segments where the player intervened only frames during the actual intervention were processed. Every frame was processed during interventions. Every 2nd frame during the entire scene with the soldier was processed in segments where the player remains passive. Overlay screens were filtered out in all segments (although these only occurred during segments where the player remained passive, not during interventions). There were 9185 samples from segments where the player intervened and 13370 samples from segments where the player remained passive.

- Second CNN test set (Test CNN 2): the second test set used to test a convolutional neural network trained via transfer learning. In segments where the player intervened only frames during the actual intervention were processed. Every frame was processed during interventions. Every 2nd frame during the entire scene with the soldier was processed in segments where the player remained passive. Overlay screens were filtered out in all segments (although these only occurred during segments where the player remained passive, not during interventions). There were 3383 samples from segments where the player intervened and 6327 samples from segments where the player remained passive.

- Choice detection test set (Test Choice): the test set used to test the method to detect which choice the player makes in a scene with the soldier based on the convolutional neural network predictions. It consisted of the same 18 segments as in the first and second CNN test sets, but here every 30th frame was processed for the entire scene with the soldier for all segments. The overlay screens were filtered out in all segments. There were 13 segments where the player intervened and 5 where the player remained passive.

6.3.2 Classifying frames

The confusion matrices of the first convolutional neural network on the training and test sets are presented in Tables 24 and 25. Its statistics are given in Table 26. As there was a significant amount of confusion between the two classes using this network, a second convolutional neural network was trained via transfer learning. Here the samples for interventions no longer consisted of the entire scene with the soldier, but only the actual interventions. The confusion matrices for this second convolutional neural network on the training and test sets are presented in Tables 27 and 28. Its statistics are given in Table 29.

		Predicted class		
		Intervene	**Passive**	**Total**
True class	**Intervene**	13458	0	13458
	Passive	0	13370	13370
	Total	13458	13370	

Table 24: Confusion matrix of the predictions on the first CNN training set of the first convolutional neural network trained via transfer learning to detect the player's choice.

		Predicted class		
		Intervene	**Passive**	**Total**
True class	**Intervene**	5198	1343	6541
	Passive	2329	3998	6327
	Total	7527	5341	

Table 25: Confusion matrix of the predictions on the first CNN test set of the first convolutional neural network trained via transfer learning to detect the player's choice.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train CNN	1	0	1	0	1	1	1
Test CNN	0.79468	0.36810	0.63190	0.20532	0.69058	0.71464	0.73898

Table 26: Results of the first convolutional neural network trained via transfer learning on the training and test sets for detecting which choice the player makes during a scene with the soldier. All results are rounded to 5 decimals.

		Predicted class		
		Intervene	**Passive**	**Total**
True class	**Intervene**	9185	0	9185
	Passive	0	13370	13370
	Total	9185	13370	

Table 27: Confusion matrix of the predictions on the second CNN training set of the second convolutional neural network trained via transfer learning to detect the player's choice.

		Predicted class		
		Intervene	**Passive**	**Total**
True class	**Intervene**	3321	62	3383
	Passive	1479	4848	6327
	Total	4800	4910	

Table 28: Confusion matrix of the predictions on the second CNN test set of the second convolutional neural network trained via transfer learning to detect the player's choice.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Train CNN 2	1	0	1	0	1	1	1
Test CNN 2	0.98167	0.23376	0.76624	0.01833	0.69189	0.84130	0.81168

Table 29: Results of the second convolutional neural network trained via transfer learning on the training and test sets for detecting which choice the player makes during a scene with the soldier. All results are rounded to 5 decimals.

6.3.3 Choice detection

The method used to determine what choice the player makes during a scene with the soldier was described in Subsubsection 5.4.2 (,which was analogous to the method described in Subsubsection 5.3.2). This method was used with the second convolutional neural network, of which the results were presented in the previous Subsubsection. It was tested on the choice detection test set. Its results are given as a confusion matrix in Table 30 and as statistics of the predictions of scenes in Table 31.

		Predicted class		
		Intervene	**Passive**	**Total**
True class	**Intervene**	13	0	13
	Passive	1	4	5
	Total	14	4	

Table 30: Confusion matrix of the predictions on the choice detection method based on convolutional neural network predictions on the choice detection test set.

	Rec	FPR	TNR	FNR	Prec	Acc	F1
Test CNN 2	1	0.2	0.8	0	0.92857	0.94444	0.96296

Table 31: Results of the choice detection method choice detection test set. All results are rounded to 5 decimals.

7 Discussion

In Sections 5 and 6 possible approaches to answer the research questions and their results are discussed. These approaches may have limitations to their usefulness or to what degree they can be generalized. It is also possible that other methods could achieve better results. These matters are discussed in this Section. This is done individually for each of the four steps taken to answer the research questions. Preprocessing is discussed in Subsection 7.1, detecting the supermarket in Subsection 7.2, detecting the scene with the soldier in Subsection 7.3 and detecting the player's choice in Subsection 7.4.

7.1 Preprocessing

Thresholds on the similarity of color histograms between frames were used to detect overlay screens. This method was used to create a list of frame numbers where overlay screens appear or disappear. This list was then used during further processing to determine whether a frame should be taken into account. I.e. it was used to determine which frames should be inserted into a training or test set for solving the later problems. Color histograms and using a list of frame numbers with a boolean to keep track of whether a frame should be processed are both computationally inexpensive methods. The downside of using a boolean to do so is that all frame numbers in the list of screen changes have to be manually verified, because a wrongfully detected screen change causes exactly the opposite frames to be processed from that frame onward until another wrongfully detected screen change is reached. While this entire preprocessing step does not affect processing videos that were not seen during this research, it is a step that would have to be reproduced when the methods developed during this research would be adapted to recognizing a different location with a different scene in ‘This War of Mine’ or in an entirely different game. The time saved by using a computationally cheap method over more complex methods to detect overlay screens is for a large part nullified by having to manually verify all detected screen changes.

An alternative would be to train an additional SVM or convolutional neural network to detect overlay screens. If an overlay screen is detected by such a method, it could be excluded from further processing and there is no longer a need for a fault-sensitive list of frame numbers where overlay screens appear or disappear. An additional benefit would be that determining where a supermarket visit ends would become easier. Currently, a threshold is used on the number of frames where the supermarket is not detected. As was mentioned earlier, this sometimes wrongly detects that a visit has ended when an overlay screen is present for a prolonged period of time. When frames can be directly classified as containing an overlay screen, this can be used to determine that a supermarket visit has not yet ended, because visits to any location in ‘This War of Mine’ cannot end with an overlay screen. The player has to actively move to the exit of a location or the part of day has to end (at the end of the day, the shelter automatically moves to the scavenge selection screen and at the end of a night a scavenging run automatically ends with the character moving back to the shelter). Since time is paused when an overlay screen is present, it follows that a visit to a location cannot end when an overlay screen is present. When the overlay screen disappears the location will appear again. If the player was at a particular location, such as the supermarket, before the overlay screen appeared, deduction can be used to determine that the player is still at that location when the overlay screen disappears again and that it is not a new visit to the same location.

7.2 Detecting the supermarket

In the previous Subsection a discussion was given about how a different method to detect overlay screens might be able to also improve the detection of supermarket visits. This method could also be beneficial for another reason, namely that the intervals used to detect the end of a supermarket visit consist of a specific number of frames, which was quite large. These large intervals were necessary to accommodate for the presence of overlay screens near the end of a scene. When overlays themselves can be detected, the window which needs to be looked at to determine the end of a scene could be reduced, which would make the method more robust.

The framerate of the videos in the dataset was usually 30 frames per second, but if videos with a higher framerate were to be presented, the thresholds would not be as effective (because the time considered would be smaller). Unfortunately, the classification results were not perfect so some kind of threshold is necessary. The window size could be adapted to the framerate of the video for which it is used to solve this potential problem.

A method such as the one described in the previous Subsection would not help with reducing the number of falsely detected scenes. These are caused by clusters of incorrectly classified frames. In general the performance of the SVM used to predict whether a frame contains the supermarket is good. However, when it classifies a frame incorrectly, it often classifies a sequence of frames incorrectly leading to these incorrectly detected supermarket visits. Temporal smoothing improved the classification results a little, but not enough to prevent falsely detected supermarket visits. To further reduce the number of falsely detected scenes the classification method would need to be improved or changed. A possible improvement would be to use spatial pyramid matching, which was used to try and improve the detection of the scene with the soldier. There, it did lead to an improvement, but the overall results were not yet satisfactory, which lead to the need for a convolutional neural network. Spatial pyramid matching could lead to a larger improvement in detecting the supermarket, because the differences between the supermarket and other locations in the game are larger than those between the different possible scenes in the supermarket. Therefore, it is possible that the spatial relationships in frames would be better captured through spatial pyramid matching, because the difference between classes is bigger. In turn, that might lead to a reduction in false positives in the frame classifications.

A different possibility would be to use a different classification method entirely. For detecting the scene with the soldier, convolutional neural networks turned out to perform better than SVMs. The same could be possible for detecting the supermarket. The downside is that a convolutional neural network is a black box method, so it would no longer be possible to inspect the decision boundary or the words in the codebook used for SVMs, reducing the interpretability of the process. If a neural network would be used, it would probably be better to construct a new neural network from scratch, instead of training a network through transfer learning, because a domain-specific network is likely to be more able to closely model the differences in the data, than adapting a network from a different domain, which starts out with features modeled on different data.

7.3 Detecting the scene with the soldier

Many approaches were tried to detect the scene with the soldier. Yet, only one of the most complex methods, convolutional neural networks, achieved satisfactory results, which shows that this is not a simple problem to solve. A convolutional neural network was trained via transfer learning from an AlexNet[5] [1] implementation. A tailor-made network would probably achieve even better results on separating the segments with and without the scene, but would also require a lot more work to set up properly. Additionally, it would be a very domain-specific network, that would likely not scale well to other problems. Although the convolutional neural network used to detect the scene with the soldier does have an increase of 10 percent of recall over the second-best method, a SVM with bag-of-visual-words histograms using spatial pyramid matching, the absolute recall is still only 78.2%. For this reason, in order to detect whether a segment contains the scene with the soldier, a threshold on the percentage of frames classified as the scene with the soldier was still necessary. The downside of a threshold is its lack of scalability. A new threshold would have to be determined for every other scene the process, that was developed as part of this research, would be adapted to. A tailor-made network may have an advantage in this regard. If the recall would become high enough with such a network, the frames classified as the scene could also be used to identify where the scene takes place, instead of the current method where a threshold is used on the entire segment to determine if that segment contains the scene with the soldier, but which does not give information about where in a supermarket visit the scene takes place.

The current process does have an advantage in its possibility to be generalized. The fact that AlexNet could be adapted to a different domain through transfer learning not only speaks of the quality of the features in its network, but it also indicates that the process of detecting a specific scene in gameplay videos of 'This War of Mine' is easier to generalize. After all, to get good results it was not necessary to build a new network from scratch (although doing so may improve the results at the expense of costing a lot of time). To detect other scenes, the same AlexNet network could be used to train new

[5] https://www.cntk.ai/Models/CNTK_Pretrained/AlexNet_ImageNet_CNTK.model

networks through transfer learning by giving it enough training data of such other scenes. The general power of convolutional neural networks means it is likely that the process can also be successfully generalized to different scenes and games.

7.4 Detecting the player's choice

The convolutional neural network that was used to detect the player's choice was trained via transfer learning from the same implementation of AlexNet[5] [1] as the network used to detect the scene with the soldier. Therefore, for the same reasons as discussed in the previous Subsubsection a tailor-made network may be able to achieve better results. Presumably also at the expense of generalizability like before. The same concerns about using thresholds to detect whether the scene with the soldier takes place, as described in the previous Subsubsection, can also be applied to the threshold used to detect which choice the player makes. Additionally, as was already mentioned in Subsubsection 5.4.1, the relatively small number of available videos with each of the choices to extract frames from for training may also have affected the results. Therefore, having more videos available that display the scene with the soldier may also be able to improve the results.

8 Conclusions

In Section 4 formal research questions were asked about each of the three sub-problems: detecting the supermarket, detecting the scene with the soldier and detecting the player's choice. Their corresponding research questions are asked in Subsections 4.2 up to and including 4.4. The formal answers to these questions are given below:

- Research question 1.1: to predict if a frame containing gameplay footage of 'This War of Mine' displays the supermarket location a SVM can be used. This SVM is trained with bag-of-visual-words histograms of SIFT features of frames displaying the supermarket against frames not displaying the supermarket. The predictions can be slightly improved by using 'temporal smoothing', which adjusts the prediction of a frame if all surrounding frames have a different predicted class.

- Research question 1.2: the start and end of a visit to the supermarket in gameplay footage of 'This War of Mine' can be determined by using the predictions about whether individual frames contain the supermarket. By finding a series of consecutive frames which are predicted to display the supermarket and using thresholds on the percentage of frames classified as displaying the supermarket, the starts and ends of supermarket visits can be determined. The results can be slightly improved by detecting loading screens at the starts and ends of supermarket visits to get more precise start and end frames.

- Research question 2: determining if a scene takes place where a soldier attempts to rape a woman in a supermarket in gameplay footage of 'This War of Mine' can be achieved by training a convolutional neural network via transfer learning to detect if a frame displaying a supermarket visit contains the scene with the soldier or a different scene. A threshold on the percentage of frames that have been predicted as displaying the scene with the soldier can then be used to determine whether the scene with the soldier takes place during a supermarket visit.

- Research question 3: determining whether the player chooses to intervene or remain passive in a video displaying a scene from 'This War of Mine' where a soldier attempts to rape a woman in a supermarket can be achieved by training a convolutional neural network via transfer learning to detect if a frame displaying this scene contains an intervention or a player that remains passive. A threshold on the percentage of frames that have been predicted as displaying an intervention can then be used to determine which choice the player made in a video containing the scene with the soldier. Detecting that the player killed both the soldier and the woman turned out to not be possible using this approach due to a shortage of videos where this choice is made.

This research has shown that the process of analyzing a scene can be largely automated. There is a need for domain knowledge to tune the various thresholds that were used, but recognizing individual frames is automated using multiple machine learning methods. The trained methods can also be used in an automated way on new samples, which were not seen during this research. To predict which choices a player makes in any new data, no additional work is required. The current system can be used to process any such data. The process developed during this research can be adapted to different scenes, which makes it a valuable step in the final goal of the overarching research project to be able to build a decision tree of the behavioral choices made by a player playing the game.

References

[1] Cntk examples: Image/classification/alexnet. https://github.com/Microsoft/CNTK/tree/master/Examples/Image/Classification/AlexNet. Retrieved: January 28, 2019.

[2] Histograms. https://docs.opencv.org/3.4.2/d6/dc7/group__imgproc__hist.html#ga994f53817d621e2e4228fc646342d386. Retrieved: September 2, 2018.

[3] The microsoft cognitive toolkit. https://www.microsoft.com/en-us/cognitive-toolkit/. Retrieved: January 28, 2019.

[4] This war of mine. http://www.thiswarofmine.com/#description. Retrieved: August 19, 2018.

[5] This war of mine. https://store.steampowered.com/app/282070/This_War_of_Mine/. Retrieved: August 19, 2018.

[6] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 510–517. Ieee, 2012.

[7] E. Apostolidis and V. Mezaris. Fast shot segmentation combining global and local visual descriptors. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6583–6587. IEEE, 2014.

[8] J. Batchelor. Games industry generated $108.4bn in revenues in 2017. https://www.gamesindustry.biz/articles/2018-01-31-games-industry-generated-usd108-4bn-in-revenues-in-2017, 2018. Retrieved: August 22, 2018.

[9] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[11] A. Canossa. Meaning in gameplay: filtering variables, defining metrics, extracting features and creating models for gameplay analysis. In *Game Analytics*, pages 255–283. Springer, 2013.

[12] S.-F. Chang and H. Sundaram. Structural and semantic analysis of video. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 2, pages 687–690. IEEE, 2000.

[13] L.-H. Chen, Y.-C. Lai, and H.-Y. M. Liao. Movie scene segmentation using background information. *Pattern Recognition*, 41(3):1056–1065, 2008.

[14] W.-H. Cheng, Y.-Y. Chuang, B.-Y. Chen, J.-L. Wu, S.-Y. Fang, Y.-T. Lin, C.-C. Hsieh, C.-M. Pan, W.-T. Chu, and M.-C. Tien. Semantic-event based analysis and segmentation of wedding ceremony videos. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 95–104. ACM, 2007.

[15] M. G. Christel and R. Yan. Merging storyboard strategies and automatic retrieval for improving interactive video search. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 486–493. ACM, 2007.

[16] W.-T. Chu and Y.-C. Chou. Event detection and highlight detection of broadcasted game videos. In *Proceedings of the 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication*, pages 1–8. ACM, 2015.

[17] W.-T. Chu and S. Situmeang. Badminton video analysis based on spatiotemporal and stroke features. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 448–451. ACM, 2017.

[18] C. Collyda, E. Apostolidis, A. Pournaras, F. Markatopoulou, V. Mezaris, and I. Patras. Videoanalysis4all: An on-line tool for the automatic fragmentation and concept-based annotation, and the interactive exploration of videos. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 470–474. ACM, 2017.

[19] S. Dasiopoulou, V. Papastathis, V. Mezaris, I. Kompatsiaris, and M. Strintzis. An ontology framework for knowledge-assisted semantic video analysis and annotation. In *Proc. 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004) at the 3rd International Semantic Web Conference (ISWC 2004)*, 2004.

[20] S. de Smale, B. van den Brink, R. C. Veltkamp, and J. T. Jeuring. Analysing player decision-making of a moral dilemma through a computer vision analysis of Youtube gameplay videos. 2017.

[21] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu. A mid-level representation framework for semantic sports video analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 33–44. ACM, 2003.

[22] M. Freire, Á. Serrano-Laguna, B. M. Iglesias, I. Martínez-Ortiz, P. Moreno-Ger, and B. Fernández-Manjón. Game learning analytics: learning analytics for serious games. In *Learning, design, and technology*, pages 1–29. Springer, 2016.

[23] Z. Gu, T. Mei, X.-S. Hua, X. Wu, and S. Li. Ems: Energy minimization based video scene segmentation. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 520–523. IEEE, 2007.

[24] C. Hentschel, I. Blümel, and H. Sack. Automatic annotation of scientific video material based on visual concept detection. In *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, page 16. ACM, 2013.

[25] C.-L. Huang, H.-C. Shih, and C.-Y. Chao. Semantic analysis of soccer video using dynamic bayesian network. *IEEE Transactions on Multimedia*, 8(4):749–760, 2006.

[26] K. Jørgensen. Audio and gameplay: An analysis of pvp battlegrounds in world of warcraft. *Game Studies*, 8(2), 2008.

[27] M. Kashif, T. M. Deserno, D. Haak, and S. Jonas. Feature description with sift, surf, brief, brisk, or freak? a general question answered for bone age assessment. *Computers in biology and medicine*, 68:67–75, 2016.

[28] N. Khan, B. McCane, and S. Mills. Better than sift? *Machine Vision and Applications*, 26(6):819–836, 2015.

[29] C. Kim and J.-N. Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 12(2):123, 2002.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[31] P. Lankoski and S. Björk. Formal analysis of gameplay. In *Game Research Methods*, pages 23–35. ETC Press, 2015.

[32] W. Lao, J. Han, and P. H. De With. Automatic video-based human motion analyzer for consumer surveillance system. *IEEE Transactions on Consumer Electronics*, 55(2), 2009.

[33] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[34] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[35] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[36] R. Marczak, J. van Vught, G. Schott, and L. E. Nacke. Feedback-based gameplay metrics: measuring player experience via automatic visual analysis. In *Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System*, page 6. ACM, 2012.

[37] F. Markatopoulou, V. Mezaris, and I. Patras. Deep multi-task learning with label correlation constraint for video concept detection. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 501–505. ACM, 2016.

[38] E. McDonald. The global games market will reach $108.9 billion in 2017 with mobile taking 42%. https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/, 2017. Retrieved: August 22, 2018.

[39] T. Mei, L.-X. Tang, J. Tang, and X.-S. Hua. Near-lossless semantic video summarization and its applications to video analysis. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(3):16, 2013.

[40] D. Milam and M. S. El Nasr. Analysis of level design'push & pull'within 21 games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pages 139–146. ACM, 2010.

[41] K. Mitgutsch and N. Alvarado. Purposeful by design?: a serious game design assessment framework. In *Proceedings of the International Conference on the foundations of digital games*, pages 121–128. ACM, 2012.

[42] P. P. Mohanta, S. K. Saha, and B. Chanda. A heuristic algorithm for video scene detection using shot cluster sequence analysis. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 464–471. ACM, 2010.

[43] M. Mühling, R. Ewerth, T. Stadelmann, B. Freisleben, R. Weber, and K. Mathiak. Semantic video analysis for psychological research on violence in computer games. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 611–618. ACM, 2007.

[44] J. Mun, P. H. Seo, I. Jung, and B. Han. Marioqa: Answering questions by watching gameplay videos. *CoRR, abs/1612.01669*, 3, 2016.

[45] A. Nantes, R. Brown, and F. Maire. A framework for the semi-automatic testing of video games. In *AIIDE*, 2008.

[46] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[47] K. Radde-Antweiler, M. Waltemathe, and X. Zeiler. Video gaming, let's plays, and religion: the relevance of researching gamevironments. *Gamevironments*, 1, 2014.

[48] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.

[49] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[50] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[51] S. Saad, S. Mahmoudi, and P. Manneback. Semantic analysis of human movements in videos. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 141–148. ACM, 2012.

[52] R. A. Sharma, V. Gandhi, V. Chari, and C. Jawahar. Automatic analysis of broadcast football videos using contextual priors. *Signal, Image and Video Processing*, 11(1):171–178, 2017.

[53] S. Siersdorfer, J. San Pedro, and M. Sanderson. Automatic video tagging using content redundancy. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 395–402. ACM, 2009.

[54] C. G. Snoek, M. Worring, J.-M. Geusebroek, D. Koelma, and F. J. Seinstra. On the surplus value of semantic video analysis beyond the key frame. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[55] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005.

[56] Y. Song, M. Zhao, J. Yagnik, and X. Wu. Taxonomic classification for web-based videos. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 871–878. IEEE, 2010.

[57] D. Strickland. Games industry to earn $108.9 billion in 2017. `https://www.tweaktown.com/news/57455/games-industry-earn-108-9-billion-2017/index.html`, 2017. Retrieved: August 22, 2018.

[58] T. H. Trojahn and R. Goularte. Video scene segmentation by improved visual shot coherence. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web*, pages 23–30. ACM, 2013.

[59] C. Wu, Y.-F. Ma, H.-J. Zhan, and Y.-Z. Zhong. Events recognition by semantic inference for sports video. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE international conference on*, volume 1, pages 805–808. IEEE, 2002.

[60] M. Yeung, B.-L. Yeo, and B. Liu. Segmentation of video by clustering and graph analysis. *Computer vision and image understanding*, 71(1):94–109, 1998.

[61] D. Zhong and S.-F. Chang. Structure analysis of sports video using domain models. In *null*, page 182. IEEE, 2001.

[62] W. Zhou, A. Vellaikal, and C. Kuo. Rule-based video classification system for basketball video indexing. In *Proceedings of the 2000 ACM workshops on Multimedia*, pages 213–216. ACM, 2000.

[63] Z. Zivkovic and F. Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.