Emotional Mario - Using Super Mario Bros. to Train Emotional Intelligent Machines

Henrik Svoren

Thesis submitted for the degree of
Master in programming and system architecture
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2020

Emotional Mario - Using Super Mario Bros. to Train Emotional Intelligent Machines

Henrik Svoren

Emotional Mario - Using Super Mario Bros. to Train Emotional Intelligent Machines

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

This thesis explores the field of *Affective Computing* and emotional intelligent machines, by implementing and investigating a specific approach for training emotional reinforcement learning agents, in the novel scenario of playing the video game Super Mario Bros.. Our approach, which is inspired by previous work in the field, is to collect specific types of data, from real humans playing the game, that correlates to emotional reactions, and then use this data to train a convolutional neural network (CNN) to predict this emotional reaction on the basis of information from the game environment. This CNN is then applied to create a custom reward scheme for a reinforcement learning agent, which introduces an *intrinsic* emotional reward, that is integrated with the pre-existing extrinsic reward of a standard deep Q network (DQN) algorithm. This thesis documents the work of (i) collecting and processing the required data into an open dataset called Toadstool, (ii) training CNN models on blood volume pulse data, to predict an emotional reaction based on frames from the game, (iii) the development of a DQN that learns to play the game and, (iv) the integration of the predicted emotional signal into the reward function of the DQN, to create an emotional DQN (EDQN). We compare the performance of various EDQN models with the standard DQN, and produce results that show that some implementations of the EDQN outperforms the DQN in certain respects.

Acknowledgments

The completion of this thesis would not have been possible without the many people who have helped me along the way. First of all, I want to thank my supervisors, Michael Riegler and Pål Halvorsen, for all of their their guidance and support, and for providing me with a fun, educational and interesting master project.

I also want to thank Vajira Thambawita, for his technical assistance and many other valuable contributions, Hanna Borgli, for her helpful advice and support, as well as all the others who either participated in the data collection, gave a hand for the dataset paper, or contributed to the thesis in some other way.

Most importantly, I want to extend a special thank you to my informal supervisor Steven Hicks, for being a consistent and reliable source of help and encouragement for me every step of the way, and for making this whole process a lot less stressful, and more enjoyable, than it otherwise could have been.

Finally, I want to take this opportunity to express my gratitude to my wonderful family and friends, for all of their support, encouragement and input over the last five years. Most of all, a very special thank you to my two favorite girls; Hanna, for making it possible, and Iselilja, for giving me the motivation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The creation of artificial entities with minds and abilities that are similar in nature to those of humans, is one of the hardest, and also potentially most rewarding, challenges facing our modern imagination, science and technology. As we shall explore in this work, one of the likely obstacles in the way of such advancements, is giving machines the ability to recognize, express and use mechanisms that are akin to human emotions. There are also many other, non-theoretical applications and systems, which could benefit greatly from an increased understanding of how to apply emotions in computing. These include social systems like certain robots or chat applications, but also other types of systems, like learning agents in the field of machine learning.

In this thesis, we will explore the challenge of utilizing emotion in computing systems, through the field of *Affective Computing*, by investigating the concept of emotional intelligent machines, and applying a specific approach for introducing emotion into the learning process of an artificial agent playing a video game. In this introductory chapter we will; discuss our initial motivation for the thesis; detail our problem statement, research question and objectives; explain the game context and framework; as well as, walk through the limitations and scope, the research methods used, and the main contributions of the thesis. At the end of the chapter, we give a brief summary of the structure of the thesis, and the contents of each chapter.

1.1 Motivation

The human imagination has had a continuing and long-standing fascination with the idea of artificial creatures and minds that have the ability to think, act, and feel in a similar way to humans. As we will expand on in later sections, this concept has ancient roots, however, the idea has never been more relevant or pressing than in modern times. Especially in view of the development and rapid advancement of computers and other related technology like robotics. The advent of these modern marvels has allowed the idea of artificial, intelligent entities to step out of the world of imagination and myth, and enter into the realm of reality. By extrapolating from

what has already been achieved, it has become easier and easier to imagine a future where such entities not only exist, but do so with a high degree of sophistication, and even play a major role in our societies.

Recent advancements in artificial intelligence (AI), specifically machine learning (ML) and deep learning (DL), have largely accelerated this development. However, they have also brought to the forefront the significant limitations of our current approaches in this regard, as well as some major hurdles that must be overcome to achieve real general purpose AI. One of these challenges has to do with emotional, or affective, qualities. Evidence has accumulated across a range of fields [2, 14, 27, 43, 63], which indicate that the pure rationality and logical abilities, which most of our current methods are based on, might be inefficient, or even insufficient, as a basis for thinking and acting autonomously in the world. Humans rely upon emotional signals and insight for many crucial cognitive processes and usually experience much difficulty functioning if these are impaired. If emotion has such a utility for humans, it might be reasonable to suspect that they could provide similar benefits to the minds of artificial entities as well.

Furthermore, humans not only wish to create AI that can think and act somewhat independently of human direction, we also have a goal of creating intelligence's that are recognizable to us as thinking entities, and that we can experience meaningful communication and interaction with. This likely requires more than a machine that is restricted to only performing logical calculations and making pure rational decisions. For such an entity to be convincing it would need to have a significant level of understanding and mastery of human-like emotional dimensions as well. This challenge is the central motivation behind the field of *Affective Computing*, which seeks to imbue machines with emotional abilities.

Many strategies have been employed to make progress in this regard [12, 48], some of which will be discussed in this thesis. One particular strategy that we found to be inspiring is the method of training deep neural networks on real human data to predict an emotional signal, that can then be integrated into an artificial agent to produce an emotional dimension. The main inspiration for the work in this thesis is the paper *Visceral Machines*, written by Daniel McDuff and Ashish Kapoor [46], where they employ such a strategy in an autonomous driving scenario. Our belief that this particular area is rich with unexplored potential, motivated our research into the field of *Affective Computing* and emotional intelligent machines.

1.2 Problem statement and objectives

Inspired by the work of McDuff and Kapoor [46], in which they used deep neural networks trained on real human data to recreate a visceral emotional response in artificial agents, we wanted to explore other areas where the same kind of principle could be applied. Specifically we decided to investigate a reinforcement learning scenario where we believed the

introduction of a human-based, internal reward-signal might improve the performance of the algorithm, or otherwise change its behaviour in interesting ways. The scenario we choose was that of playing video games, or more specifically, playing some genre-representative, specific game that could serve as an entry point for research into this area. Thus, the main research question we are trying to answer over the course of this thesis is;

> *Can human emotions be used to improve the performance, or otherwise change the behaviour, of an autonomous agent playing video games?*

To answer this question, we break it down into three objectives that, when completed, will bring us closer to a final conclusion. The three objectives will be completed in order, with each building on the achievement of the objective that came before it. In the following, we describe each objective in detail.

As with any supervised learning task, we need the right kind of labeled data in order to train the machine learning algorithms that would predict the human signal. That is, we needed data on some analysable, emotionally correlated signal of humans playing a video game, as well as the associated frames, and other data, from the game itself. To the best of our knowledge no such dataset existed that was publicly available, and so we decided that it would be a good idea to create one. Therefore, the first part of this thesis concerns the objective of;

Objective 1 Develop a dataset for the purposes of conducting research involving the physical state, signals and behaviour, related to emotional states, of humans playing a video game. The dataset should contain; game data from human play sessions, correlated physiological and/or other observational data, and other information relevant to analysing the data.

This dataset, while applicable to a range of research areas, will also give us the elements needed to attempt to recreate an emotional signal from human subjects that can be integrated into a reinforcement learning agent. The second part of this thesis will attempt to;

Objective 2 Train a deep neural network to predict an emotionally correlated human signal based on the collected dataset.

This neural network will serve as the provider of the emotional signal that shapes the intrinsic reward in a reinforcement learning agent. The third and final part of this thesis will therefore focus on the following objective:

Objective 3 Build a model of a reinforcement learning agent that plays the same video game and perform experiments to show how its learning and behaviour is affected by introducing the emotional signal as an intrinsic reward mechanism.

By introducing the emotional signal to the reinforcement learning agent, and running experiments on the resulting models, we aim explore the potential utility of this approach for reinforcement learning in games, and for emotional artificial intelligence in general.

1.3 Game context and framework

Figure 1.1: Start screen of the game Super Mario Bros. First released on the Nintendo Entertainment System in 1985.

In order to develop a dataset of the kind described in the previous section, we first needed to decide on a game that would serve as our specific environment for the project. We decided to go with Super Mario Bros. (SMB) as the game to use in the study and the following experiments. Choosing this game had several perceived advantages. Firstly, SMB is a well known and much played title, with a game formula that makes it easy to find participants with varying levels of experience. Secondly, it is relatively simple in terms of graphics and gameplay, which makes the analysis easier to fit within the scope of a master thesis. Thirdly, there are existing frameworks for emulating and developing reinforcement learning models on this game. This meant that we could use a pre-existing reinforcement learning environment, which saves us the work of having to build one from scratch.

Framework

The framework we chose to use is the gym-super-mario-bros environment [37], which is a OpenAI Gym [8] based environment. OpenAI Gym is a toolkit for working with reinforcement learning in a wide range of different simulated environments. The gym-super-mario-bros framework builds on this to create a setting for reinforcement learning using SMB. This framework seemed ideal as it could easily be modified for human play sessions,

as well as being already optimized for later reinforcement learning experiments.

1.4 Limitations and Scope

Grounded in the stated objectives, the scope of this thesis entails;

1. The collection and creation of a game specific dataset for human emotional signals.
2. The training of deep neural networks to predict human emotional reactions.
3. Conducting experiments comparing a standard reinforcement learning agent, with decent performance in the environment, to the same agent while including the emotional signal as an intrinsic reward.

Within each of these areas, there is a plethora of possibilities to explore. However, due to time limitations, as well as the amount of work required for each part of this thesis, we are forced to limit the scope of our project to exploring a narrow subset of these areas.

Firstly, there are a multitude of ways in which we can collect data to be used for our experiments. In our project, we limit ourselves to exploring a single game, collecting game, video and physiological data on 10 participants in this setting. This gives us a good range of participants, but is not an adequate participant pool to be truly representative of any group. The dataset is restricted to the specific physiological measurements available through the E4 Empatica wristband monitor, all though there are certainly many other kinds of measurements that could be interesting in this context. Additionally, many lessons were learned during and after the collection of the dataset which could have been applied to generate an even more useful data collection, there was however no time for this as the first data collection was itself quite a lengthy process.

There are also several of the data categories collected that could be used, exclusively or in combination, for predicting an emotional reaction. In this thesis we only explore one way to leverage the blood volume pulse (BVP) signal to predict an emotional response. Moreover, a more stringent preprocessing of the BVP signal, including a more customized peak detection algorithm, would have been desirable. The same can be said for the process of training the predictive networks, as more time would have given the opportunity to construct a more effective model.

Finally, due to resource limitations, we explore a certain type of *intrinsic* reward implementation, in a specific implementation of a double deep Q network reinforcement learning algorithm, playing the first level of the game, using one emotional predictive model. All these factors could potentially be adjusted for further exploration. We do test a model for each participant's predictive model, as well as test several different combinations of intrinsic and extrinsic rewards for certain participants.

As stated, there are many possibilities and variations on many aspects of this process that could potentially yield interesting results, but will

remain outside the scope of this thesis. We discuss some of these, as well as other potential ideas for future work, in the chapter 7.

1.5 Research Methods

For any work involving scientific research, it is important to have a theoretical framework of the research methodology within the field. There are various such theoretical frameworks that can be used as a foundation for research. This thesis is somewhat interdisciplinary, touching upon fields like psychology and neurology, however, it remains fundamentally a project of computer science. Therefore, it should also be theoretically rooted in this field in terms of research methodology.

We have chosen to use the Association for Computing Machinery's (ACM) as a research methodology for this thesis. In 1989, a task force was assigned by the ACM Education Board, to create a report detailing the core fundamentals of computer science and engineering [17]. In this report they envision computing as consisting of three distinct paradigms; theory, abstraction and design. These paradigms are described as intricately intertwined and inseparable, yet distinct in that they represent different areas of competence. The work in this thesis thus relates to all of these paradigms in different ways. In this section we will briefly discuss each of the paradigms, and mention certain instances of how they relate to our project.

1.5.1 Theory

The *theory* paradigm is taken from the field of mathematics and relates to the development of a coherent, valid theory, and describing the relationships among objects. There are four steps that are applied to achieve this. These steps are iterated over when errors or inconsistencies are found;

1. Definition: characterize objects of study
2. Theorem: Hypothesize possible relationships between them
3. Proof: Determine weather the relationships are true
4. Interpretation of results

An example of how we use this paradigm in our thesis, is in the analysis of the relationships between the parameters of the DQN reinforcement learning agent, and its performance and training efficiency in the game environment. We theorize that the performance of the DQN is sensitive to changes in certain parameter values. Adjusting these might improve learning performance for the agent and increase the efficiency of the algorithm. We run experiments to show how the performance is affected by changes . Based on our interpretation of the results, we then run new experiments to further optimize the algorithm.

1.5.2 Abstraction

The *abstraction* paradigm, which can also be described as the *modeling* or *experimentation* paradigm, is rooted in the experimental scientific method and relates to the investigation of a phenomenon, and to making predictions about the world in the form of a hypothesis. Four stages are iterated over in this paradigm;

1. Form a hypothesis
2. Construct a model and make a prediction
3. Design an experiment and collect data
4. Analyze results

One major way in which this paradigm can be identified in this thesis is in the neural networks trained to predict BVP amplitudes, and the behaviour and training performance of the reinforcement learning agent. We hypothesize that the predictive networks might improve learning performance for the agent when included in the value function. We construct a model to test, and predict that the agent will achieve better scores and finish rates with fewer training cycles. We run experiments to collect data on how the performance is affected. Based on our analysis of the results, we then run new experiments with a more specified hypothesis to further explore the relationships we uncover.

1.5.3 Design

The *design* paradigm is rooted in the engineering and relates to the construction of systems or devices to solve specific problems, or perform other useful actions. This paradigm also consists of four steps that are iterated over until the system meets the given requirements;

1. State requirements
2. State specifications
3. Design and implement the system
4. Test the system

A good example of the use of this paradigm can be observed in our development of the Toadstool dataset, and especially in the process of constructing the data collection tools and protocol needed for this. The nature of the experiments we ultimately wanted to run meant that we had to collect certain types of data, which largely set the initial requirements for the data collection system. Further requirements were built upon this groundwork, considering scope, participants, framework etc. Based on these requirements, we set certain specifications for building the system. The system was then designed and implemented according to the specifications. Testing of the system revealed certain weaknesses that lead to the adoption of new requirements, which lead to redesigning the system, and so on.

1.6 Main Contributions

Over the course of writing this thesis, we have done research in the area of emotional intelligent machines and *Affective Computing*, as well as produced tools and data to support this research. Our work has been aimed specifically at training deep convolutional neural networks (CNNs) to predict an emotional signal based on real human data, and applying these to a deep Q network (DQN) doing reinforcement learning in a game setting. Our work should be viewed as a preliminary exploration of this idea, as there are many further improvements and alternative paths of inquiry that could have been prioritized. We will here restate the objectives set in Section 1.2, and discuss our main contributions in association with each of them.

Objective 1 Develop a dataset for the purposes of conducting research involving the physical state, signals and behaviour related to emotional states, of humans playing a video game. The dataset should contain; game data from human play sessions, correlated physiological and/or other observational data, and other information relevant to analysing the data.

To meet this objective, we created a dataset called Toadstool [70], which contains game, video and physiological data from sessions where a real human player plays the game SMB. We also built the tools, protocol and methods required to collect, process, reproduce and extend the data collection. The dataset has the necessary data to perform experiments of the kind described for our context, as well as additional data that may well make it interesting for other types of research as well.

Objective 2 Train a deep neural network to predict an emotionally correlated human signal based on the collected dataset.

This objective is supported by the training of CNN to predict BVP amplitudes from the collected and processed BVP data in the Toadstool dataset. We showed that this can be achieved to a certain degree through a simple pipeline, all though there is still much room for improvement here in order to make even more accurate predictions.

Objective 3 Build a model of a reinforcement learning agent that plays the same video game and perform experiments to show how its learning and behaviour is affected by introducing the emotional signal as an intrinsic reward mechanism.

The final objective is supported by our creation of an emotional DQN model, that uses the emotional signal provided by the previously discussed neural network, to create an intrinsic reward signal for the reinforcement learning agent. We conduct experiments testing models for each of the participants in the dataset. We also conduct experiments putting varying emphasis on the emotional signal, as well as with a decaying emphasis. We show some promising results that indicate that there might be some learning benefit to the emotional signal.

1.7 Thesis Outline

This thesis consists of seven chapters, including this introductory one. The second chapter contains the background information necessary for understanding the context, motivation and work in the thesis. Chapter 3, 4, 5 and 6 describes the various work performed during the course of this project. Chapter 7 is the conclusive chapter, in which we summarize and discuss the work and results produced, as well as lessons learned and possible future work. In this section we give a brief summary of chapters 2 to 7.

Chapter 2: Artificial Intelligence, Emotions and Reinforcement Learning

This chapter contains background information on the various fields explored in this thesis. We discuss the broad concept of emotional intelligent machines in history and popular culture. We also take a look at machine learning techniques and deep neural networks, including CNNs. Also included is a section briefly discussing the most relevant psychological theories of emotion. Finally we discuss the field of *Affective Computing*, and why we should be interested in giving emotion to intelligent machines, as well as how this can be achieved. In this context we touch upon methods for affect recognition and expression, intrinsic emotions and emotions in the field of reinforcement learning.

Chapter 3: Collecting the Toadstool Dataset

In this chapter we present, and explain the process behind developing, the Toadstool dataset. This dataset contains game, video and physiological data from 10 participant playing SMB We discuss various aspects of how we planned and prepared the study, including the setup for the game session, the study protocol and the software tools and equipment used for collecting the data. We also discuss the actual collection process and the participants, as well as the synchronization done in post-processing. In conclusion we present the resulting dataset, and consider some possibilities of how it can be applied.

Chapter 4: Simulating Emotion using the Blood Volume Pulse

This chapter is dedicated to the process of training CNNs, using the BVP data from the Toadstool dataset, in order to reproduce a signal representing a human emotional response to situations in the game. We discuss the method of photoplethysmography for capturing BVP, and explain the connection between BVP and emotional arousal caused by sympathetic nervous system responses. We show how we processed the raw BVP data into normalized amplitudes suitable for representing such a response. Finally, we explain the process of training CNN models to predict the BVP amplitudes for each participant.

Chapter 5: Building a Double Deep Q-learning Network to play Super Mario Bros

In this chapter we discuss how we constructed a double deep Q network (DDQN) based reinforcement learning agent to play SMB This agent will serve as the foundation for building models that integrate an emotional signal, as well as a performance baseline for testing these models. We detail the environment, including the base environment and the various wrapper functions used, as well as the base reward function. We also detail the neural networks used in the model. Finally, we test the performance of the DDQN in the given game setting, and adjust certain parameters (learning rate, memory and image quality) to improve the performance of the model according to our purposes

Chapter 6: Introducing Emotion into the DQN

In this chapter we use the CNN's trained in chapter 4 to create an intrinsic, emotional signal, which is integrated into the reward function of the base DQN algorithm to create an emotional DQN (EDQN). We explain the theoretical framework that we apply for introducing emotion. We then detail how we shape and integrate the predicted BVP amplitudes into the DQN. Finally we conduct experiments where we test and compare the training performance of different EDQN agents. We train one agent for each participant in the dataset with a certain emotional emphasis in relation to the extrinsic reward. We also do experiments with varying, as well as decaying, emotional emphasis. Finally we make some observations and discuss conclusions that can be drawn from the experiments.

Chapter 7: Conclusion and Discussion

In this final chapter we look back on and discuss the work done in this thesis, make concluding remarks, and suggest possibilities for future work. We begin with summary off the previous chapters, and continue with a reiteration of the objectives set in the introductory chapter and how we have met each of them in turn. We conclude with a discussion of lessons learned during the project, as well as limitations and suggestions for possible future improvements and work. In this section we focus on the areas of data collection and processing, training and evaluation of the CNN models, and the implementation and testing of the EDQNs.

Chapter 2

Artificial Intelligence, Emotions and Reinforcement Learning

In this chapter we will explore several subjects that are relevant to understand the positioning of this thesis, as well as the work performed in the later chapters. In the first two sections we will briefly discuss how the concept of emotional intelligent machines has evolved through human history. We will then have a section with an overview of the most relevant psychological theories of emotion, before moving on to discuss machine learning and other AI related topics. Finally we explore why and how to develop emotional intelligent machines, in the context of Affective Computing. We look at affect recognition and expression, having emotions and using emotions in reinforcement learning, as well as the *Visceral Machines* [46] project.

2.1 Early Concepts of Emotional Intelligent Machines

The idea of humans creating artificial creatures possessing a level of intelligence similar to our own, which often also possess an internal emotional component, to serve as servants or companions, is a very old one in the human imagination, and the concept is well represented in our lore and cultures. For instance, the ancient Greek philosopher Aristotle imagined that if we were ever to produce instruments that could work themselves and understand the wishes of humans, it would lead to the end of slavery and servitude;

> *For if every instrument could accomplish its own work, obeying or anticipating the will of others, like the statues of Daedalus, or the tripods of Hephaestus, which, says the poet, "of their own accord entered the assembly of the Gods"; if, in like manner, the shuttle would weave and the plectrum touch the lyre, chief workmen would not want servants, nor masters slaves.* — Aristotle [5]

Other examples of this idea, found in ancient Greek mythology, is in the form of legends that depict dragons teeth that are sowed into the ground and subsequently grow into ferocious warriors called *the spartoi*. Most prominent of these legend is perhaps the story of Cadmus, who killed a sacred dragon and, after growing its teeth into warriors, had them fight each other to the death over a precious jewel. He later lead the survivors to found the city of Thebes [54].

Another early example of this idea is the legend of the *golem* from Jewish folklore. These were anthropomorphic creatures, made from clay, mud or other inanimate matter, that came to life through magical rituals [34]. Golems were often depicted as perfectly obedient servants, but with a tendency to interpret their masters too literally. In these ancient depictions, we can already see the intuition emerge that such artificial servants will benefit from possessing something more than simply the ability to understand language and to move and operate independently. That is, they need some kind of internal, emotional dimension and understanding. In the case of the golem, if they had a better understanding of the emotional content of their masters instructions, they would not so easily misinterpret their intentions. In the case of the spartoi, these were not merely mindless drones, but possessed the capability of being independently motivated to act based on self-interest, as evidenced by Cadmus' ability to make them fight to the death over a valuable jewel.

In later centuries, the belief that humans and animals are no more than complex, biological machines, and so could be artificially created, emotions and all, if sufficiently understood, has been held by many thinkers and cultures throughout our history. In western culture, one of the most successful proponents of such a view was the 17th-century philosopher Renè Descartes. Although personally claiming a special divinity and emotional dimension for humans, Descartes believed that the human biology and brain, as well as all animals, were mechanistic in nature [30]. Descartes' behaviourist view, as well as his other work in philosophy and science, has been hugely influential on much of later western thought. Without his claim of the esoteric divinity of human emotion however, there is no reason to assume that the mechanistic nature of animal behaviour, does not extend to human capabilities as well. The idea that living beings are no more than complicated machines supports the idea of us being able to understand, as well as recreate, even the mental and emotional capabilities of humans and animals. Descartes' *language test* was likely also a critical inspiration for Alan Turing when he created the first idea of the Turing test [1], the famous thought experiment that aims to determine whether a computer is capable of exhibiting intelligence that is indistinguishable from that of a human.

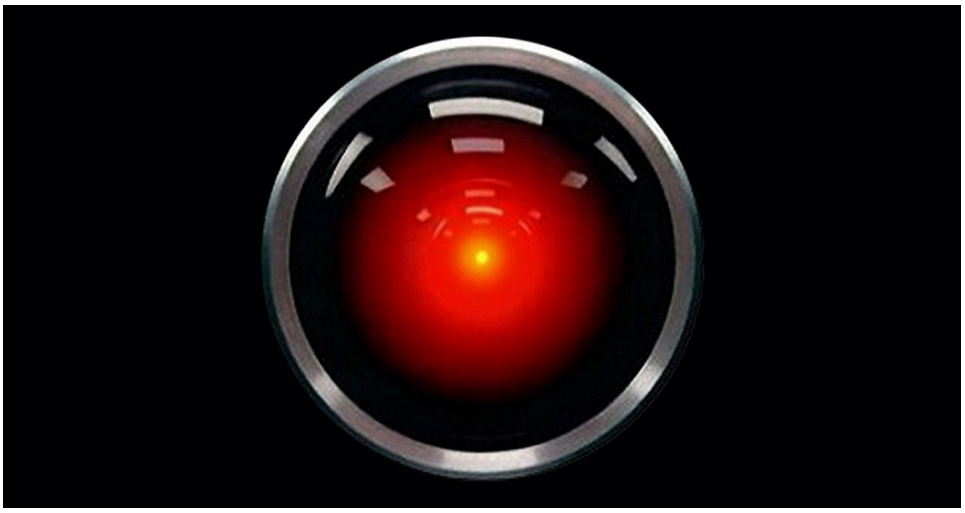

Figure 2.1: The interface of the intelligent on-board computer HAL 9000, from the movie *2001: A Space Odyssey* (1969).

2.2 Emotional Intelligent Machines in Popular Culture

The idea of emotional intelligent machines, as well as its encompassing idea of artificial general intelligence, has had many varying representations within modern products of the human imagination. The popularity of science fiction media has given us plenty of imagined examples of artificial entities possessing emotion and intelligence, and how such entities might be useful, or detrimental, to human endeavours.

One of the most iconic of these characters is the computer HAL 9000 from Stanley Kubricks *2001: A Space Odyssey*. In this movie from 1969, HAL 9000, the onboard computer on a spaceship, displays a wide variety of emotional abilities including emotional speech, recognizing emotions and social skills, in addition to the logic and computational capabilities expected from a supercomputer. Over the course of the movie we can see HAL go from being a productive asset to the crew, to the antagonist of the story, through a combination of challenging situations and emotionally influenced reasoning.

In HAL, although fictional, we can see the alluring promise of what emotional intelligent AI might become, as well as an idea of the potential dangers of such intelligence. One such danger is the potential for internal emotional states to trigger unexpected behaviours in the machine that might misalign the goals of the AI with those of humans, such as HAL killing crew members in an effort of self-preservation. Another is the danger of emotional intelligent machines manipulating humans through an understanding of human affective processes. Both of these elements are present in this quote taken from the peak of the movie when Dave, the human operator, is shutting HAL 9000 down, and the computer appeals to

their shared emotionality as it pleads for its life; "Stop Dave. Stop Dave. I am afraid. I am afraid Dave."

HAL represents the classical idea of artificial intelligence that develops emotionality as a by-product of its other intellectual capabilities and the nature of its tasks. The onboard computer is a disembodied entity, aside from the embodiment in the ship itself, that is designed to manage technical systems and serve as a predictable and helpful companion with a limited range of social skills. The complexity of HALs systems eventually gives rise to an internal emotionality that is unpredictable, and explicitly self-serving. In this, we recognize HALs internal states as closer to our own, however, these capabilities were not a designed feature of the system, but rather a “bug” that was a consequence of HALs ability for self-awareness and reflective thought. Whether self awareness and emotion can arise from the underlying complexity of systems in this way remains an open question. As we will discuss in later chapters of this thesis, logical and rational capabilities might not suffice as a basis for such capabilities. Still, HAL, and similar imagined intelligences, have played a huge role as an inspiration for research into artificially intelligent systems and the development of emotional abilities in machines.

Another, more contemporary, well known depiction of emotional intelligent machines, can be seen in the robot Ava, from the 2014 movie *Ex Machina*, written and directed by Alex Garland. Ava, unlike HAL, is embodied in a human-like form, seemingly experiencing the world in a similar fashion to humans, and has been explicitly designed with sophisticated emotional abilities. This movie presents a different vision of AI, where emotions, perspective, and inherent goals are an integral part of the artificial agent. Creatures like Ava introduce us to another dimension of the potential of AI, where these entities go beyond being mere autonomous servants and become true physical and emotional companions.

Figure 2.2: The emotionally intelligent robot Ava, featured in the movie *Ex Machina* (2014).

Ava also shows us some of the dangers associated with this technology. As machines and robots, like Ava, gain increased emotional understanding and become more adept at imitating nuanced human behaviour, the potential for them to be used for manipulation and other nefarious purposes also become more pressing. If AI should develop its own goals and purposes, that might be totally inscrutable from the human perspective, these abilities could become truly problematic. Such a scenario plays out in *Ex Machina*, where Ava is a captive of her creator and, due to her curiosity and instinct for self-preservation, has a hidden goal of escaping her situation. She uses her superior intellect and understanding of human psychology to manipulate the protagonist, a programmer sent to test her abilities, into helping her achieve this. She expertly elicits emotions like empathy, and convinces her subject of their emotional connection. A connection which, in the end, is proven to be only part of a calculating strategy on Ava's part, as she casually abandons her saviour to a terrifying fate.

As emotions in humans are seen as inherently unstable, unpredictable and often irrational, there will likely arise issues like this that future implementations of emotional intelligent AI will have to contend with, as we approach more sophisticated implementations of this technology. However, despite the warnings presented in these fictional accounts of AI, such risks are unlikely to deter the human quest to create emotional intelligent artificial entities. The potential rewards and benefits of such technology are simply too attractive, and the potential risks seem distant, as well as largely in the realm of science-fiction at the current stage of technological development. In fact, even if the risks were considered a major concern, the gains to be had from both general AI, as well as emotional AI, are just too obvious and consequential to expect humans to not pursue this technology. Especially as advancements in research continues to chase our imaginations of AI, gradually bringing entities like HAL and Ava into the realm of what is possible.

2.3 Theories of Emotion

As much of the work in this thesis is concerned with emotions and affect, in the context of computing and machine learning, a quick recap on the major psychological models of emotions that have been applied in this context, seem in order. There are several different psychological models of emotion that have been applied in computing contexts. Sometimes these are in conflict, but they can also be somewhat complementary, as they focus on different aspects of emotional processes. In this chapter we will briefly explore the three most dominant theories of this kind; categorical, dimensional and componential theories.

2.3.1 Categorical emotions

Most of the literature that is concerned with emotion and affect in the context of machine learning, employs categorical models [48]. Categorical theories assumes that there are basic emotions that can be sorted into discrete categories.

Perhaps the most influential work in this vein is that of Ekman, in which he identified six 'basic' emotions that are universally recognized across cultures; anger, fear, joy, sadness, surprise and disgust [19]. Each distinct emotion can be considered as an action tendency that has been shaped by evolutionary processes. For instance, feelings of joy has the action tendency of approach, of wanting to move towards and making contact with the source of the emotion, while fear is associated with the action tendency of avoidance, prompting us to stay away from from dangerous situations [21].

Even tough the categorical perspective has been very influential, the existence of fundamental emotional categories remains a controversial claim in psychology. Many variations of categorical models exists, that include anywhere from 2 to 18 different emotions, underscoring the uncertainty in the field over exactly what emotions should be included in such a model [48]

2.3.2 Dimensional emotions

Another approach to emotion that is well represented in computer related research is dimensional theories. As early as 1897, the great psychologist Wilhelm Max Wundt suggested that emotions could be understood along the three dimensions of pleasurable/unpleasurable, arousing/subduing and straining/relaxing, laying the foundation for a dimensional approach to emotion [83].

Instead of discrete emotional categories, dimensional theories assumes the existence of an affective space, within which emotional states may be identified along different dimensions. Proponents of this approach argue that categorical models fail to explain our intuitive perceptions of differences and similarities between emotional states, like the intuition that some emotions seem to share certain characteristics but not others, or that certain feeling are experienced as opposites [69].

Several dimensional strategies have been proposed with a varying number of dimensions, however the two most widely accepted dimensions are valence, the degree to which an emotional state is negative or positive, and arousal, which refers to levels of activation elicited by an emotion [60]. Perhaps the most influential model of this kind is the circumplex model, developed by James Russel, where emotions can be located on a circular map with two axis, representing valence and arousal [59]. An illustration of this model can be seen in Figure 2.3. A criticism that dimensional theories are often faced with is that they can have trouble distinguishing between certain types of emotions that are similar to each other, such as anger and disgust.

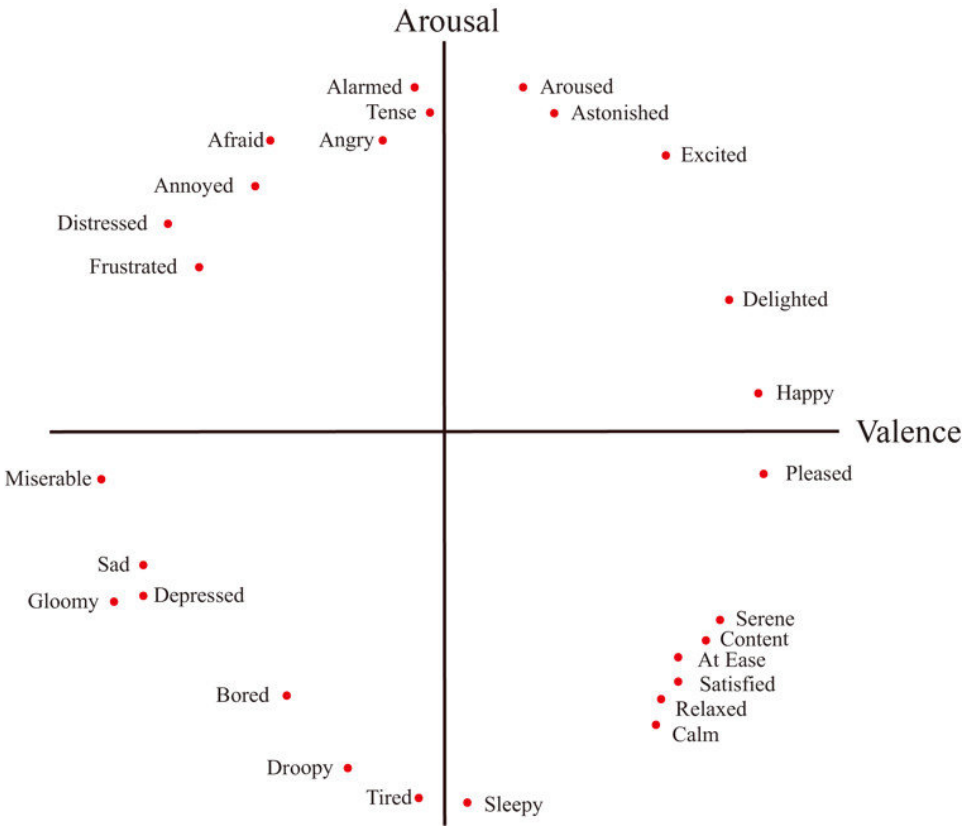

Figure 2.3: Russel's circumplex model of emotion [59]

2.3.3 Appraisal Theory and Componential Emotions

A third approach that has sometimes been applied in computational frameworks is componential emotion theory, also known as cognitive appraisal theory [40]. This view focus on how emotions are elicited by external factors, and considers emotions to be the result of evaluations, or appraisals, made on the basis of how some stimuli is personally relevant. For instance, fear might be the result of appraising a situation as potentially harmful to our person or goals. Some appraisal categories that are often used are valence, novelty, goal relevance and coping potential. Componential theory provides a nice description of the structure of emotion, but has been criticised for being unable to give an account of how the appraisals actually arise or evolve, and how emotion functions in relation to other cognition.

2.4 Machine Learning and Reinforcement Learning

Machine learning is considered to be a sub-field of AI and refers to a set of techniques that allow computers to "learn from experience". By feeding data through statistical models that identify patterns, these techniques enables the machine to gradually figure out how to improve its performance on certain complex tasks, without the need for any task-

specific programming. The field is neatly summarized in a quote from Tom A. Mitchell;

> *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.* — Tom A. Mitchell [47]

Machine learning has in recent decades seen a huge growth in popularity and success, largely due to general increases in processing power that allows researchers to more easily run complex models. There are already many applications that employ these techniques to great success. Some of these include web searches, filters, recommendation systems, ad placement, credit scoring, fraud detection, stock trading and many others. Thus, machine learning has already significantly impacted many aspects of society, and is predicted to impact many more in the future.

One area that is currently being revolutionised by machine learning techniques is medicine. For instance, machine learning is set to drastically improve prognosis, as well as diagnosis, by being able to consider many more variables, in a much shorter time frame, than a human doctor could possibly do manually [49]. Machine learning techniques will also have an immediate impact on medical work that largely consists of analyzing digitized images, like that of radiologists [10]. While we will still need human radiologists at the wheel for the foreseeable future, machine learning techniques are showing that when it comes to tasks like image classification, it will be a tall order for humans to compete with the future performance of machines, both in terms of efficacy and efficiency.

Much of the research discussed in this thesis fall within the area of machine learning. In addition, the models and experiments we produce also rely on techniques of this type. Machine learning methods are often divided int three main categories; supervised learning, unsupervised learning and reinforcement learning. In this section we will briefly explore these in turn, paying particular attention to the field of reinforcement learning and deep Q models.

2.4.1 Supervised Learning

Supervised learning is a process in which an algorithm is trained on data that has been labeled. This means that the desired output associated with each sample in the training data is known. The training process goes like this; When given a sample, the algorithm tries to predict the correct output. It then adjusts its internal weights based on how close its prediction was to the desired output. This process is repeated with each new sample, letting the algorithm gradually improve with each iteration. The process ends when the algorithm stops improving, the data runs out or some other specified threshold is reached.

Common types of of supervised learning include classification algorithms, where the output is a discrete category, and regression algorithms, where the output is a value within a range. Supervised learning is used for

a number of applications like image and speech recognition, spam filtering, credit scoring and many others. Some popular models of this type are decision trees, support vector machines and neural networks.

2.4.2 Unsupervised Learning

In unsupervised learning scenarios, the training data is unlabeled. Since the desired output is unknown, the algorithm has only the inputs to work with and so cannot check to see if its predictions are correct. Instead, it attempts to find patterns within the unlabeled data to perform some task.

A very common use for these types of methods is cluster analysis, where the algorithm groups data points into clusters, based on patterns of similarity. Some common clustering techniques are K-means clustering, which tries to minimize the distance between some data point to the average data point of the cluster, and hierarchical clustering, which continually merges clusters that are close to each other. These techniques can be used, for instance, to sort, or otherwise pre-process, large sets of unlabeled data, or for applications like spam filters, recommendation systems [51] or even for the detection of fake news [32].

Unsupervised learning has also been applied to create generative models. Generative models aim to learn how to create new data that is similar to the training data. By identifying the patterns and characteristics of the training data, such models are able to generate new samples that could plausibly have been a part of the training dataset. A recent generative framework that has gained a lot of popularity is generative adversarial networks (GAN) [25]. This method engages two neural networks in a *contest* against each other. The *generative network* tries to create new, plausible data, while the *discriminative network* tries to tell the generated samples apart from the original data. GANs have been used successfully for a wide variety of applications, some of which include; the creation of new art [4], scaling up of graphics and visuals of video games [80], as well as the creation of new medicinal drugs [41].

2.4.3 Reinforcement learning

A special class of machine learning algorithms are those that belong to the category of reinforcement learning. Reinforcement learning can not be strictly said to be supervised learning, as the algorithm does not learn from labeled data, nor does it satisfy the definition of unsupervised learning, as the target output of the algorithm is usually known in advance. Reinforcement learning frameworks operate by placing an agent into an environment and having it learn from its experiences. The input to the model is the current state of the environment, while the output is the behaviour, or actions, of the agent, with the desired output simply being the desired behaviour. These models learn by the help of a reward function which assigns a positive or negative reward to each action. The agent adjusts its internal weights to reflect these rewards and so gradually

improves its ability to predict actions that will give better results, moving towards an optimal policy.

A reinforcement learning system that has made headlines in recent years is the AlphaGo program, which was the first to achieve superhuman performance in the game of Go, consistently defeating the human Go champion Fan Hui in a series of games in 2015 [65]. All-tough artificial intelligence had already surpassed humans in many tasks and games, this was viewed as a particularly groundbreaking achievement because of the enormous space of potential moves and possible games in Go. This version of the AlphaGo program, used a combination of supervised learning, where the program was trained on data from expert Go-players. and reinforcement learning, for finding optimal policies. A later version of the program, called AlphGo Zero, showed that it could achieve even higher levels of performance without using supervised learning or any previous human knowledge [66]. This version of the program begins from a "blank slate" and learns by way of pure reinforcement learning, playing against itself and evaluating its own games. The resulting AlphaGo Zero was able to beat the previously mentioned version 100-0 in a series of games.

From basic Q-Learning to Double Deep Q-Learning Networks

Q learning is a simple reinforcement learning algorithm, dating back to 1992, that learns a policy which tells an agent what action it should take in any particular situation [81]. The algorithm fills out and updates a table of which actions to choose in all possible states. The algorithm incorporates the temporal dimension of learning by predicting an action based on its cumulative, potential future reward. The importance of future rewards is determined by a discount factor between 0 and 1, where low values will make the agent focus more on short-term rewards, and higher values will shift more weight onto long term rewards. Similarly a learning rate is set at a decimal fraction which determines the degree to which new knowledge overrides the old information.

A further development of the Q-learning algorithm is what is often referred to as deep Q-learning. The term "deep" refers to the use of deep learning techniques, discussed more in section 2.5. This method is able to handle far more complex environments, where the set of all possible states becomes very large, by replacing the predicting table with a neural network that predicts the best action to take in the current state. The introduction of neural networks opens up many possibilities, but also leads to a certain instabilities in the predicting function, due to things like sequential observations often being too similar and small changes to Q resulting in large changes in data distribution.

To address this, the deep Q method adds the technique of 'experience replay' to the framework, which means that instead of simply learning from the most recent action taken, the program randomly picks a batch of samples from an index of previous actions and their correlated states and rewards, also called memories, to learn from at any given learning step. The data structure used to store the information about previous memories

is called the replay buffer, and is a type of dynamic memory that, when at capacity, will overwrite the oldest information in favor of new memories. This technique removes correlations in the data and evens out changes in the data distribution. The resulting algorithm is commonly referred to as a DQN, an abbreviation for Deep Q-learning Network.

Another improvement to this approach came in the form of what is called Double Deep Q-learning Networks (DDQN) [73]. The standard DQN algorithm struggles to perform in certain environments due to a tendency to overestimate the value of certain actions. This happens because the same Q-function is used for both estimating the maximum value of current actions, and for estimating the maximum future value of actions [29]. A solution to this is introduced by the DDQN method, which uses two separate neural networks, one for selecting the current action and another for estimating the maximum value of future actions. The network that predicts future values, is only periodically updated to ensure that target future values largely remains stable during training. The resulting DDQN algorithm is shown to outperform the standard DQN in many environments, particularly stochastic ones, and has become the standard for most implementations of DQNs. For the rest of this thesis, when we use the term DQN, we will be referring to the DDQN version.

2.5 Artificial Neural Networks and Deep Learning

Artificial neural networks (ANNs), or simply neural networks, are a specific type of computational model that has roots in the neurological structure that we find in the animal and human brain. All though originally inspired by neurology, neural networks have since become cemented more as a engineering principle, which achieves superior performance for certain machine learning problems. Like humans, neural networks learn by way of studying examples. Unlike humans however, they usually need a lot more of them to learn anything useful. For instance, where a human could learn to accurately separate between two types of images, say of cats and dogs, with just a few examples of each, a neural network might need thousands, or even millions, of such examples before achieving the same accuracy. As neural networks, and CNNs in particular, are integral to the work done in this thesis, this section will discuss how these models work on a basic level and how they are related to the concept of *deep learning*.

2.5.1 Deep Learning

Deep learning refers to a broad class of machine learning algorithms which are closely connected to the computational model of ANNs. In essence, any neural network with enough internal layers can be considered *deep learning*, but exactly how many layers are needed for a network to be "deep" is a somewhat fuzzy subject. However, most consider neural networks with more than a couple of layers to represent deep learning. In this thesis

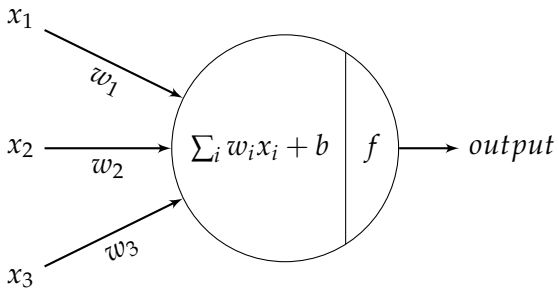

Figure 2.4: A visualization of the artificial neuron, or the Perceptron. It takes three weighted inputs, passes them through an activation function, and produces a single output.

we employ deep learning in the form of CNNs, which are used both for predicting the BVP signal, and in the DQN used to play SMB.

2.5.2 Artificial Neurons

The fundamental units of neural networks are called artificial neurons, or simply neurons. Modern neural networks typically consists of tens or hundreds of thousands, or even millions of individual neurons, working in tandem to calculate solutions to a given problem. The individual properties of the neurons are simple, but in large collections they are able to deal with very complex calculations. A visualization of the neuron can be seen in Figure 2.4.

A typical neuron takes a undefined number of weighted inputs. The neuron calculates the weighted sum of the inputs and adds a bias term. The bias terms is typically 1 or -1, and has the effect of adjusting the output in a positive or negative direction. Then, in order to add an element of non-linearity to the process, the result is passed through what is called an activation function. Finally, the output of the neuron is passed to a neuron in the next layer of the network, or, if its the final layer, given as the output of the model. The activation function is crucial to the effectiveness neural network, as it is the element that enables networks of multiple neurons to function better than a single one. Many different activation functions have been proposed and used since the arrival of neural networks.

2.5.3 Multilayer Perceptrons

A good way to get a deeper understanding of how neurons and neural networks function, is to look at one of the earliest successful implementations of neural networks, which was called the Perceptron. Frank Rosenblatt, on the back of the work of Warren McCulloch and Walter Pitts, developed this model in the mid-20th century [55]. The Perceptron is a binary linear classifier, structured like the neuron in Figure 2.4. Its activation function consists of a threshold algorithm which activates if the sum of its inputs are greater than zero (the function is expressed in equation 2.1).

$$f = \begin{cases} 0 & \text{if } \sum_i w_i x_i + b \leq 0 \\ 1 & \text{if } \sum_i w_i x_i + b > 0 \end{cases} \tag{2.1}$$

While Perceptrons showed much early promise, they where in time revealed to have some crucial limitations. The main problem is that the Perceptron can not solve problems that are not linearly separable. This means that it is only able to distinguish between categories that can be separated by a straight line. This is often expressed as the XOR problem, as one its manifestations is that the Percoptron cannot perform the XOR logical operation. Even though a single perceptron can easily learn logical operations like AND and OR, it is impossible to teach it the XOR operation.

Luckily, a solution to the XOR problem was discovered, by combining several Perceptron neurons in a multilayer network. A simple network of a first layer with two neurons, and a second layer with a single neuron, is able to perform the XOR operation. A three layered network can approximate any function according to the universal approximation theorem [11]. This type of network is called a multilayer perceptron (MLP) and is considered the most basic type of neural network. An illustration of this structure can be seen in Figure 2.5.

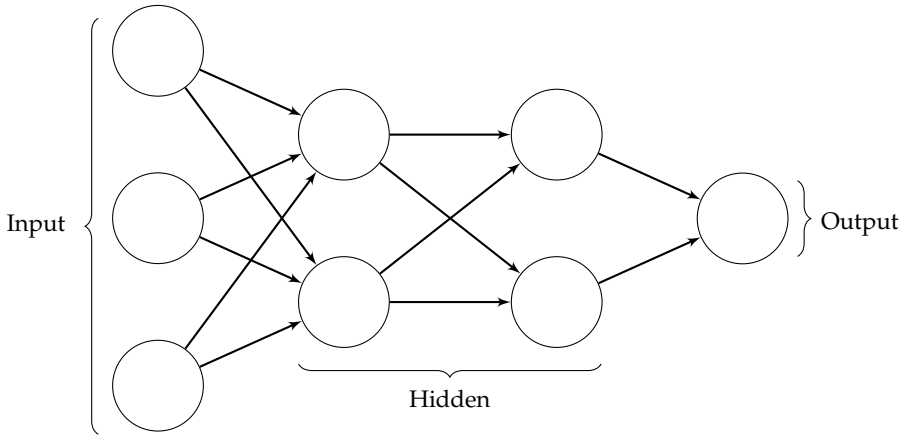

Figure 2.5: A visualization of the layers of a multilayer perceptron. There are three types of layers; input, hidden and output.

We can see that each layer belongs to one of three categories, depending on their placement in the order of layers. The first layer is called the input layer, and is simply designed to receive the initial data that is given to the network, and pass it forward to the next layer. This layer performs no computations and contains no adjustable weights for learning, and is therefore excluded when referencing the actual depth of a network.

The final layer is the output layer, which produces the output of the network. The specifications of this layer are usually specific to the given problem to be solved and its domain. In the case of a network used for classification, this layer usually the final classifier. A common function for classification is *softmax*, which contains a neuron for each class, and assigns

a probability for each based on the input. For regression problems there is often a single neuron, which calculates the final predictions of the network in the form of a linear value.

All layers between the input and output layers are referred to as hidden layers. This is mostly due to the lack of interpretability and transparency, from the perspective of the programmer, of the calculations and parameter adjustments that happens here. The data is passed through the hidden layers, which then learn by adjusting their weight parameters, and the model makes a prediction based on its learned knowledge.

The MLP is what is known as a feed-forward neural network, meaning there is a unidirectional data flow, from input layer, through the hidden layers, and out the output layer, which does not allow for data to be passed "backwards" through layers. This is contrasted with recurrent neural networks, which allows for considering previous decisions in the neuron.

2.5.4 Training Neural Networks

Now that we have seen how neural networks are structured, we are ready to discuss how these networks are able to learn from the data they are given. The mechanism used for this are the networks weighted parameters that are associated with the connections between neurons in adjacent layers. These weights are continuously updated as the network learns from new data, further optimizing its performance. There are several different methods that can be used for updating the weights of a neural network, some of which we will look at in this section.

The most common way to apply learning to neural networks is through the technique of backpropagation [58]. This essentially entails letting the network make a prediction based on its current weights, and using a loss function to see how wrong the prediction is, before moving backwards through the network, updating the weights to more closely match the desired output.

The loss function is crucial for neural networks to learn to make better predictions from data, as it is our mechanism to gain a measure of the difference between predicted output and the ground truth. For regular classification problems, where the decision boundary is large, the most common loss function for modern neural networks is called cross-entropy. For cases where decision boundaries are small, and for regression problems, the most used loss-function calculates the mean squared error.

Once we have a loss-function, we also need a function for minimizing the loss by way of adjusting the weights of the network through backpropagation. This is often referred to as the optimization function. Most optimization functions work on the principle of gradient descent. Gradient descent is a method for minimizing an objective function by updating the parameters of a model in the opposite direction of the gradient of the function [57]. The technique comes in three variants that uses different amounts of data to calculate the gradient of the objective function; batch, stochastic and mini-batch gradient descent. These each make a trade-off between the

accuracy of parameter updates and time to perform an update, based on the amount of data used.

Batch gradient descent is the *vanilla* version, where the gradient of the objective function is computed for the entire dataset for every update. This process guarantees a stable convergence, but can be very slow and is not viable for data sets that are too large for the memory. Stochastic gradient decent takes an opposite approach, updating the weights for each training sample. This makes for much quicker process, by removing redundancies, and can also enable online learning, but also introduces some potential instability into the convergence of the model. Mini-batch gradient descent is a compromise between the two other methods, which updates the weights for every mini-batch of n training samples. This method has generally produced the best results and is the most commonly used approach to training neural networks.

There are still challenges however, with the mini-batch gradient descent method as is, especially with scheduling an appropriate learning rate. Therefore, popular contemporary optimization functions use extended algorithms to overcome these challenges. Some of the most used optimization functions include; Adegrad, Adadelta, RMSprop, Adam, AdaMax and Nadam. In this thesis we will use the Adam (Adaptive Moment Estimation) optimizer for our neural networks, which computes adaptive learning rates for each parameter.

2.5.5 Convolutional Neural Networks

The neural networks used in this thesis are all of a particular kind, namely CNNs. CNNs are similar to MLPs in many ways, but with some key extended functionalities that make them particularly powerful for many types of problems. Since they first arrived on the scene around 1990, CNNs have seen many technical improvements and have become one of the most useful tools we have in the field of machine learning. Its unique features have helped CNNs to achieve superior performance on a range of tasks, and none more so than in the field of computer vision and image classification. This is evidenced, for example, by their dominance in competitions like the Large Scale Visual Recognition Challenge (ILSVRC), for over a decade.

The main difference between the CNN, and more traditional neural networks like the MLP, is that it arranges its neurons in a grid-like topology. This makes the CNN structure particularly suited to process multidimensional data like the pixel values for an image, allowing for effective representation of images, and the recognition of patterns directly from raw-pixels without much preprocessing [28].

CNN architectures are typically composed of three types of layers; convolutional, pooling and fully-connected layers. Fully-connected, or dense, layers are simply regular neural network layers with a given number of neurons, and is usually used as the final layers to shape the output. The true innovation of the CNN architecture lies in its ability to represent large multi-dimensional data, like images, while keeping the

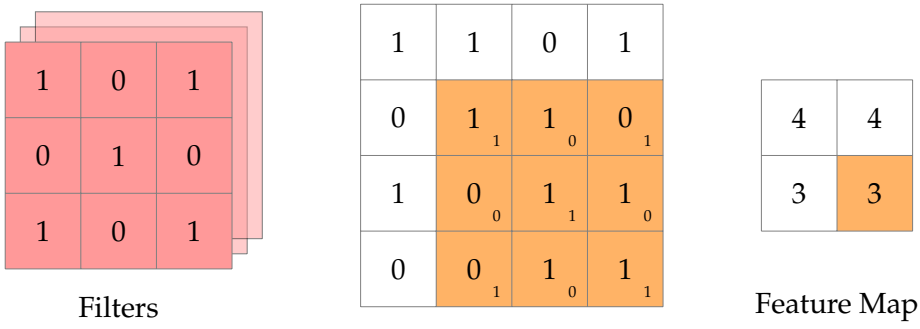

Figure 2.6: An example of a convolutional operation on a $4 \times 4 \times 1$ image using a kernel size of $3 \times 3 \times 1$ and a stride of 1.

number of parameters low, as compared to traditional neural networks, by its application of convolutional and pooling layers.

Convolutional layers

The central contribution of the CNN architecture is the grid-like topology of the convolutional layers and the convolution operation performed here. The convolutional layer tries to learn feature representations, which are called feature maps, of the inputs, which are given as data in a dimensional form. This significantly reduces the complexity of the network in terms of the amount of weighted connections needed between layers.

The parameters to this layer is a collection of learnable filters, often referred to as kernels, that come with a set size. In a regular 2-dimensional convolution the size can simply be viewed as height and width. During the forward pass of the data, the filters slide across the inputs and performs a dot product calculation between the filter and the current position of the input. This produces feature maps, which are abstract representations of the input. This process is illustrated in Figure 2.6 producing a 2×2 feature map, by using a filter of size $3 \times 3 \times 1$, applied on a input with the dimensions $4 \times 4 \times 1$.

Pooling layers

Another central feature of CNN's are pooling layers, which are most often placed in-between the convolutional layers in a CNN model. These further reduce the complexity of the network and the number of weight parameters needed. The pooling operation is similar to the convolutional one, in that it slides a window, the size of which is called the pool size, across the input. The values in each pooled are transformed into a single value, reducing the size of the data. The two most common forms of pooling operations are max pooling, which takes the maximum value from each pool, and average pooling, which takes the average of all the values in a pool. An illustration of these two pooling operations can be seen in Figure 2.7

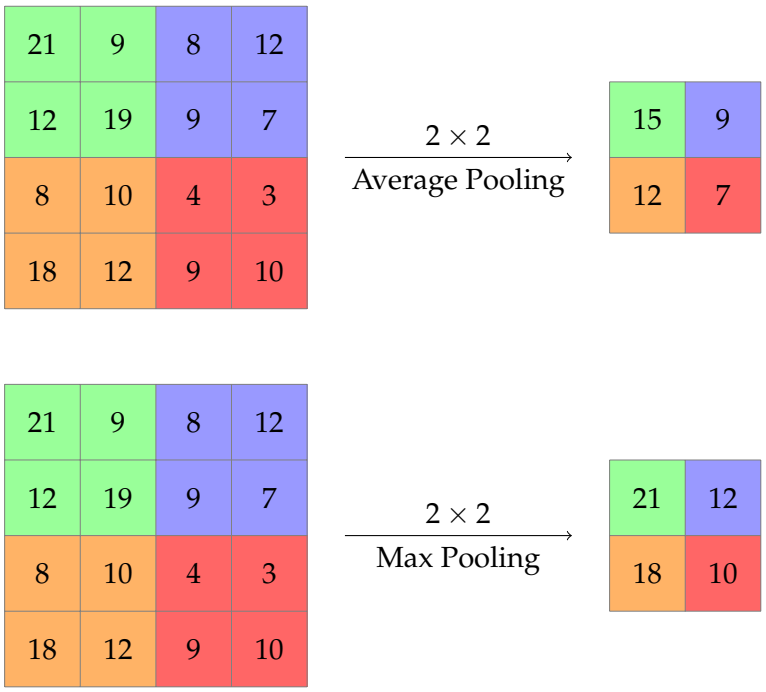

Figure 2.7: An example of the two most common pooling operations; Max Pooling and Average Pooling.

2.6 Affective Computing: Why build Emotional Intelligent Machines?

The study of *Affective Computing* concerns matters that have to do with enabling machines and devices to effectively apply emotion in various ways. In her seminal work from 1997, Rosalind W. Picard establishes the field, and argues for why we should endeavor to give machines emotional abilities [52].

Picard notes that although many thinkers believe that the paradigms of logic and rationality is all that is needed to create artificial intelligence, despite massive increases in these capabilities, scientists still struggle to produce machines that interact intelligently with humans or reason intelligently about many types of problems. She cites insights from neuroscience, cognitive science and psychology that indicate the crucial role of emotions in reasoning and problem solving, motivation, perception, cognition, coping, creativity, attention, planning, learning, memory and decision making, to argue that producing such machines will require them having a robust set of emotional abilities as well.

There is indeed a sizable amount of literature across several fields of study that support the notion that purely rational decision making is often not the optimal way for humans to think and is, in fact, not the way the human mind works in practice. For instance, findings in neurology show that intuition and emotion is an essential element in human decision making, and is intertwined with what we perceive as rationality [14].

While earlier scientists has largely considered the rational mind to be the control center of the body, in which decisions were reached through mostly undisturbed reasoning, it is now a widely accepted belief that cognition is an embodied process. That is to say, thinking and decision making is heavily influenced by the physical feelings and emotions of the body, and in fact, humans have a hard time making any decent decision without the aid of such influences [13].

One example that support this idea is neurological research which suggests that the "framing effect", the tendency of people to perceive the same options differently based on how they are presented, is particularly associated with activity in the amygdala part of the brain [16]. The amygdala is closely related to the processing of emotions, as evidenced by the observation that damage to the amygdala causes impairment of people's emotional abilities [2]. This supports the idea that emotional processes in the brain must be a part of our models of how humans make decisions. It has also been shown that affect has a special and significant role in decision making in risky and stressful situations, and that emotional reactions in these situations significantly diverges from our more rational considerations [43].

In the field of moral psychology, much focus has historically been put on the role of reasoning. However, more recent findings indicate that emotion and affective intuition are far more influential on our moral decisions [27]. Although there is a role for reason to play as well, especially when our emotional intuitions collide [26], on the whole, humans seem to base their moral decisions more on 'how they feel' about something, than on 'what they think'. If the goal is to create artificial systems that think and act in the manner of humans, or even have recognizable moral capacities, then these insights strongly suggest that emotional and intuitive abilities will have to be an integral part of such a system.

From the perspective of sociology, rational choice theory is the idea that all, or most of, human intentional behaviour is the result of rational choices, based on conditioning from different rewards and punishment systems, in which emotions play little or no part. This theory has been heavily criticized for failing to account for central aspects of human life, like collective action and social norms [63]. The preferences derived from our values and personalities are crucial determinants of our individual behaviours in rational choice theory, and yet, rational choice theory struggles to explain how such values originate and are formed in the first place [31].

Non-rational human behaviour, like someone going out of their way to help someone in need, even when it brings no foreseeable benefit to yourself, happens everyday in our societies and is by most people viewed as an important part of what makes someone "a good person". Purely rational agents however, would likely not be able establish what we as humans would consider *good* values like altruism, on their own. This supports the idea that values based on emotional states play a major role in producing behaviours that are beneficial on a collective scale, and that if

we want to create artificial agents with such *good* values, some emotional component will have to be involved.

2.7 What is required to build Emotional Intelligent Machines?

Picard imagined her own version of the Turing test for an affective computer, in which to pass, the computer would have to convince the tester of its emotional nature, as well as its intelligence. She suggests that, in order to build emotional intelligent machines that could pass such a test, we at a minimum need to imbue these machines with the ability to recognize, respond to, express, and even to have their own emotions, in the sense of regulating and utilizing emotions in making decision making[52]. In this section we will look at some of the progress made by researchers in each of these areas, before taking a closer look at the application of emotion in reinforcement learning.

2.7.1 Affect recognition

The last decades has seen an increasing interest and success in developing methods for affect recognition, the ability to automatically recognize human emotions, across several modalities and application areas [50]. The prevalent approach to affect recognition is the sensing of affect through low-level signals, either individually or in combination. Such signals include facial activity, posture, gestures, hand tension, vocal activity, text and electrodermal activity [12].

For facial activity, the facial action coding system (FACS) is a method that has been successful in helping to automatically identify and categorize facial expressions from humans. FACS works by identifying and analysing the action units involved in facial movements. The emotion facial action coding system (EMFACS) is an application of FACS that selectively scores only the action units that might be related to emotions, and so can be applied to automatically recognize and categorize human emotions from their faces [18].

Postures, the position of the body, and gestures, movement expressions with parts of the body, also express emotions and can be used to recognize affect. Several types of algorithms have been developed for detecting affect from posture and gestures, including 3D-model based, skeletal-based, which analyses the skeletal positioning, and appearance based algorithms, that extract information directly from video or images based on trained templates [12]. There are many interesting approaches and works in this area that demonstrate the promise of these modalities. For instance, one particular study that had children playing chess with a robot cat, found that they could accurately evaluate the levels of engagement with the robot, through patterns of postural behaviour[61].

Language, both spoken and written, also carry affective content that may be automatically detected. When language is vocalized, the

exact way words are spoken contain cues as to what words carry the most significance, as well as to the affective state of the speaker [22]. There has been great progress in the area of automatically detecting emotion from recorded speech, as well as towards developing applications that can do this in real-time [76]. Automatic detection of emotional content from written language is also an important area, as many interfaces rely on text-based communications and input. There are several software applications made for sentiment analysis that use natural language processing, computational linguistics and text analysis to sense emotional content from text [12].

Another method for detecting affect is through the measurement of physiological signals like electrodermal activity (EDA). EDA measures the electrical conductivity on the surface of the skin, which is the result of sweat produced by a reaction in the sympathetic nervous system that indicates arousal [15]. Such direct measurements can be very informative, however, a key limitation of EDA measurements is the inability to separate between positive and negative affect, as well their vulnerability to interference from other factors like motion.

2.7.2 Affect expression

In addition to the ability to recognize emotions, for them to be perceived as truly emotional agents, emotional intelligent machines will also need the ability to express emotions in a convincing and effective way. Here we take a look at two kinds of implementations of affect expression; those using disembodied agents, like chat-bots, and those using embodied agents, like social robots.

The first type of affect expression is through done through disembodied agents that exist as software only, which mainly operate using speech or text based communication. In recent years intelligent personal assistant conversational systems, like Apple's Siri [68] and Amazon's Alexa [3], have become increasingly popular and widespread, and are able to answer many questions and assist users in many practical tasks. These systems may incorporate some simple affective expression in their speech patterns, but are generally very limited in their ability to understand and express affect and in establishing emotional connections with users.

Some have tried to overcome this limitation by developing social chat-bots, that have a high emotional intelligence and are more geared towards serving as actual companions, meeting the users need for communication, affection and social belonging, rather than being mere assistants [64]. These systems track emotional content and changes during conversations and adjust their output in order to establish an emotional connection with the user. An example is Microsoft's social chat-bot XiaoIce. Humans that have interacted with XiaoIce have reported significant psychological benefits and the system has been successful in establishing an emotional connection with its users. With continuous improvements over time, the XiaoIce system has shown an increasing ability to recognize users emotions and

Figure 2.8: The social robot Paro interacting with an elderly user.

intent, as well as respond to their conversational needs, even during longer interactions [85].

Another type of affect expression, that comes with some additional challenges as well as opportunities, is using emotional intelligent systems that are physically embodied, like social robots. Research show that the embodiment of an emotional actor deeply influences how we perceive that actor and make us more likely to view them as an independent social entity [35]. Social robots have shown success in areas like elder care. For instance, the robotic seal Paro is designed for long-term therapeutic interaction, and has been shown to have positive effects on elderly in care homes by strengthening social ties and reducing stress [77–79]. The utility of social robots have also been demonstrated in relation to children with autism, where they can be helpful in diagnosis, as well as serve a therapeutic and teaching role by acting as a playmate, behaviour eliciting agent and social actor, that may be less intimidating and complex than actual human actors [9]

2.7.3 Machines having emotions

The final feature that Picard sees as necessary for building truly emotional intelligent machines is to enable them to have their own emotions, in the sense that they utilize emotion-like processes in their decision making. It is important to note that this can be achieved without solving the harder problem of creating machines that have an actual, internal *experience* of emotional states, if such a thing is even possible. Rather, we can aim to create artificial agents that have internal states and processes that correlates to and emulates the emotional qualities of humans, which they then apply in their decision making process. Current, low-level applications of this concept might utilize some aspect of human emotionality to aid in

improving its performance on some specific task, like some of the examples we mention in this section. However, this method, as our ability to simulate human-like emotional states improves, might also eventually lead towards artificial agents that have more complex emotional intelligence, incorporating many or most of the central aspects of emotion in a single system.

In this thesis we focus on the application of emotional factors in machine learning, and specifically reinforcement learning, scenarios, where much work has been done in this respect. There are, however, also other approaches to building agents that incorporate emotion, including using symbolic representations of emotion. For instance, the Cathexis model [74] incorporates Ekman's six emotional categories into a symbolic emotional architecture in the pet robot named Yuppy. This work was later expanded upon to create the social robot Kismet [7]. Several other symbolic emotional architectures have been developed. Although these have been successful in solving a number of tasks, they are severely limited in regards to learning from their environment and other unstructured problems [48]. This is exactly what machine learning excels at by having the agent gradually learn from exploring its environment and adapting to its given objectives.

2.7.4 Emotion in Reinforcement Learning

There has been multiple strategies applied in order to incorporate emotions in reinforcement learning scenarios. Here we will differentiate and give examples of four main classes of techniques that are applied for this purpose, namely; homeostatic, appraisal, value and reward, and hardwired methods. We then discuss the method used in *Visceral Machines* [46], which is the main inspiration for the work in this thesis.

Homeostatic methods

One type of strategy is employed by what can be called homeostatic methods. This class of techniques focus on internal drives, that are intensified or reduced as homeostatic variables change. Changes in homeostatic variables may happen as a result of the intrinsic nature of the variable, or due to some environmental factor. The idea of internal drives originates from Drive Reduction Theory, dating all the way back to 1943, which points to drive reduction as being a critical component of how humans learn [33]. A typical example of a homeostatic variable is the current sugar/energy level [48]. This variable has a temporal dynamic, as it will decrease over time, as well as an internal drive associated with low levels, in the form of hunger. This internal drive will encourage the agent to seek out ways to satiate it in the environment. In the case of humans this would be by consuming food, while artificial agents might look for other ways to increase its energy levels, like charging at a power outlet.

One example of a homeostatic reinforcement learning method can be seen in the work of Gadanho and Halla [23]. They found that, for their learning robot, introducing artificial emotions was useful in helping it

make decisions in an unstructured environment while having multiple goals. Their model is focused on categorical emotions, and includes the homeostatic variables of hunger, pain, restlessness, temperature, eating, smell, warmth and proximity. These variables are tied to emotions. For instance, the robot will become 'sad' if it has low energy, or become more fearful if it is experiencing pain.

Appraisal methods

Another class of strategies focus on applying appraisal theory to give rise to emotion that forms the basis for an intrinsic motivation. An intrinsic motivation is contrasted with an extrinsic motivation, as a reward that is based on the behaviour itself, as opposed to being a consequence of the behaviour, and has been shown to achieve improved task achievement over non-intrinsically motivated agents [67]. Appraisals here provide an affective meaning to stimuli from the environment. Appraisal dimensions that are often used include; novelty, recency, control, pleasantness and motivational relevance [48].

This approach has been successfully applied in several scenarios. For instance, Kim and Kwon [38] used cognitive appraisal theory to improve human-robot interaction, by building a robot with intrinsic emotions. Yu et al. [84] also used a this approach to increase cooperation between agents by endowing them intrinsic emotional capabilities, like the ability to recognize social fairness, that helped them learn cooperative behaviours.

Value and Reward methods

A third type of strategy to create emotion in reinforcement learning agents, focus on modifying the value and reward functions. One technique derives emotion directly from the state value, like in the work of Matsude et. al [44], which uses a separate value function to represent fear. The fear emotion, in this case, is increased when the agent receives a negative reward.

Hardwired

In addition to the previously discussed methods for eliciting emotions in reinforcement learning agents, there is a final category of applying hard-wired connections between input and emotions. In this strategy there is usually no need for additional internal processing from the agent as the expected emotion is pre-specified. This approach can be viewed as a more theoretically simplistic one, as models of this kind does not necessarily have to incorporate an understanding of complex emotional dynamics. Many implementations of this type overlap with the other three categories in various ways.

Visceral Machines

One recent contribution in the area of equipping reinforcement learning algorithms with internal emotional states, which we found to be particu-

larly exciting, is the work of McDuff and Kapoor [46], detailed in their paper *Visceral Machines: Risk-Aversion in Reinforcement Learning with Intrinsic Physiological Rewards*.

Here they explore how a self-driving, reinforcement learning system, in a simulated environment, is affected by introducing a signal that predicts the visceral emotional responses of the human sympathetic nervous system through the BVP. They collected physiological data on the BVP of human drivers that were exposed to driving sessions, and used this data to train a machine learning algorithm to reproduce the signal based on frames from the simulator environment.

The BVP of humans rises quickly when exposed to potentially dangerous situations. Leveraging the likely connection between changes in BVP and dangerous situations in the driving simulator, the researchers successfully mitigated the problem of sparse rewards in the environment, and increased the efficiency of the learning process as well as the driving performance This was achieved by integrating the reproduced BVP signal into to the reward function of the reinforcement learning algorithm creating an intrinsic, emotion-based reward. In this way the researchers successfully demonstrates the potential for emotional signals collected from humans to be used in a reinforcement learning scenario.

McDuff and Kapoor [46] also potentially shows us how a reinforcement learning agent's emotional personality can be regulated. In their experiments they adjust a variable in their code that makes the system place more, or less, weight on the recreated BVP signal. When there was more importance put on the signal the agent became more cautious, while when there was less so, the agent became more comfortable with taking risks. This makes sense since the BVP signal is correlated with potentially risky situations.

This method is in a way a type of hard-wired strategy, as the connections between the input, in this case the images from the driving simulator, and the elicited emotion, is based on the real human data and pre-determined by the trained neural networks. It is however, more flexible than many other such strategies in terms of dealing with novel situations. It also has something of appraisal based strategies, as the neural network can be viewed as generating an emotional state based on its appraisal of the current environment.

By using a deep neural network, that is trained on real human data, to reproduce a human signal that forms the basis for giving the algorithm an internal, emotional state, this approach somewhat circumvents the challenging problem of understanding how emotions are created and processed in the human brain and mind. Even though McDuff and Kapoor are explicit in that they are not attempting to mimic biological processes, at least the possibility of eventually applying such a method to give human-like emotional abilities to artificial agents is implicit in their work. Of course, such an approach is a far way off from having such an agent "feel" in the sense that humans do, having an actual experience of feeling, nor does it come close to an actual model of human biology. Still, by focusing on simply reliably reproducing the signals produced by humans' emotional

states, and connecting these to a reward system, we are able to produce internal states in an artificial agent that correlate to these human emotions, without the need for complex emotional models.

Such a method may seem to oversimplify the problem of emotion generation, and yet it may represent a quicker path towards emotional intelligent machines that have close to human emotions, than waiting for a more complete, neurological understanding of how emotions are created in the brain, or trying to build artificial models of complicated and incomplete theories of emotion. If we can successfully apply this principle to reinforcement learning agents, and we improve our ability to predict and recreate signals that are correlated with human affect, we might eventually be able to produce and regulate desired, human-like emotional states of a more complex nature in other artificial agents as well. This could lead to being able to produce systems that satisfy Picard's requirement for emotional intelligent machines to "have" emotions, in the sense that they regulate their emotions and utilize them in their decision making [52]

Chapter 3

Data Collection Study for the Toadstool Dataset

This chapter details the development of the Toadstool dataset [70], which contains game, video and physiological data from a group of participants playing SMB. The dataset paper was published at the 11th ACM Multimedia Systems Conference, and is openly available for research purposes. The main motivation for creating this dataset was to collect the data necessary to train a deep neural network to predict an emotional response based on frames from the game. We also believe that the dataset can be a useful resource for researchers involved in various types of projects. We therefore decided to make the selection of collected data as broad as conveniently possible, in order to make a richer resulting dataset for future and other work. This chapter discusses the planning and execution of the data collection process, the synchronization of the collected data, and the resulting dataset.

3.1 Planning the Study

In this section we discuss the planning and preparation for the data collection study. In order to facilitate a smooth data collection and result, many things had to be set up correctly beforehand. We will here go through the work done to ensure a successful study, including; setting up the gaming session, the capture of game data, the scoring system, the capturing of video and physiological data, the equipment, questionnaire and consent form used, the physical setup and pre-trial testing, as well as the protocol used in the study.

3.1.1 Game session setup and capture

An important requirement for the study was the ability to capture the relevant data from the game sessions of the participants. The gym-super-mario-bros framework uses the nes-py emulator[1], which is based on the

[1] https://github.com/Kautenja/nes-py

SimpleNES emulator[2]. By making some adjustments to the script used by the nes-py emulator to play as a human from the keyboard in the gym-super-mario-bros environment, we were able to modify the system so that we could play in the environment as a human while recording the information we needed from the play-session.

To record the game session itself, it was sufficient to store a list of each action, or lack of action, performed at each step of the game-loop. As the SMB environment is deterministic, such a list of actions can be used to recreate a game session at a later time. This was deemed preferable to extracting the frames during the play session itself, as images of each frame can then be extracted later at our convenience, without disturbing the processing of the game session with heavy disk operations. Another benefit of doing this is that it drastically reduces the size of the resulting dataset, as the total size of all the frames from a game session can be quite large. The reduced size makes storage, transfer and sharing of the dataset much more convenient. Furthermore, gym-super-mario-bros offer four different graphic environments to play the game in. These include the standard graphics, as well as three different down-sampled versions (down-sample, pixel, and rectangle). While the data in our dataset is collected in the standard environment, since the actions are recorded, sessions may be replayed in any of the other environments if preferable. The beginning and end of each session was marked with a UNIX timestamp stored with the actions, so that it could later be synchronized with data captured through other channels.

The experience of playing in the gym-super-mario-bros environment has some distinct differences from playing in the original *Super Mario Bros.* game released by the Japanese gaming company Nintendo in 1985. In the gym environment there is no music, nor is there any of the sound effects that usually accompany events in the game. All game-freezing animations and cut-scenes, such as transitions between levels, traveling through pipes etc., have also been removed from the game. While the gym-super-mario-bros framework offers setups for playing that are designed for reinforcement learning contexts, like replaying from a single level, over and over, with only a few lives, these are not very suitable for the kind of human play sessions we needed for the data collection. Therefore, we built a custom setup for playing the game that seemed more appropriate for our purposes by making some additional modifications to the code. Specifically, we used the gym-super-mario-bros option for playing a single level at a time as a base and created custom methods for transitioning from one level to the next.

In the session, players are tasked with completing as many levels as possible within the time given, with no limit on lives. If they die in the game, they are transported to the beginning of the level. If they complete the level they go to the next one and so on.To ensure that players would not end up stuck on a single level they were unable to complete, we implemented a time limit so that if a player died after spending a certain

[2]https://github.com/amhndu/SimpleNES

Figure 3.1: Frames are taken from each of the 32 levels contained within Super Mario Bros. Note that each image is taken from the very first frame of each level. Levels in *Super Mario Bros.* are organized in groups of four and called worlds, so the first level is world 1-1, the second level is world 1-2, the fifth level is world 2-1, etc.

amount of time (about 4 minutes and 30 seconds) on the same level, they will restart on the next one.

The varying skill level of players meant that we could not know in advance how many levels each participant would be able to complete, and therefore play, during the course of their session, beyond the seven or so levels that is required by our setup due to the time restrictions described above. Therefore, in order to ensure that we were able to collect adequate samples from various situations in the game, the levels played were set up in a specific order so that the players would be more likely to encounter a broad range of situations during their playthrough. The custom order is somewhat similar to that of the original game, but with some key differences.

The SMB game contains 8 worlds, each consisting of four levels, for a total of 32 standard levels. Examples of frames taken from each of these levels can be seen in Figure 3.1. There are several types of levels; overworld, underground, underwater, athletic (platforms) and castle, each with different situations and characteristics. The first five levels the players face are therefore each a representation of one of these types of levels. For simplicity these are just the first examples of such levels that appear in the game.

There are also some rare enemies in the game, namely; Hammer-Bros, that throws hammers in arcs, and Lakitu, which flies around the top of the screen, throwing spiky enemies at Mario. The 6th and 7th level in the ordered setup were therefore set to the first levels where these enemies show up in the game. The subsequent levels given are the skipped levels up to this point, followed by the rest of the game's levels, in their sequential order.

Order:	World	Stage
1	1	1
2	1	2
3	1	3
4	2	2
5	1	4
6	3	1
7	4	1
8	2	1
9	2	3
10	2	4
11	3	2
12	3	3
13	3	4
14	4	2
15	4	3
16	4	4
.	.	.
.	.	.
32	8	4

Table 3.1: The custom stage order used in the data collection.

The customized order of the levels that was used for the data collection can be seen in Table 3.1. By ordering the levels in this specific sequence we hope to increase our chances of collecting the appropriate data for all the different, relevant game-states.

Other significant features to note in the final setup the are;

1. Participants play for a set amount of time
2. There is no limit on lives in the game
3. The levels are played in the custom order
4. Power-ups do not carry over to subsequent levels
5. Warp zones are off-limits (players must be informed of this if necessary)

We also adjusted the parameters of the image viewer so that players would get a more appropriately sized game-window.

3.1.2 Scoring system

We included a basic scoring system for players in order to introduce a competitive element to the gaming sessions, and as a simplistic measure of performance. The idea was to make the players feel motivated to clear levels in as few tries as possible without dying, and also to be more emotionally invested in their performance. The scoring system is designed

Figure 3.2: The sensors of the E4 Empatica Wristband.

to reward clearing levels, and to penalize dying. Each cleared level gives a base score of 1000. This number is reduced, by 100, for every death, down to a minimum of 200. Therefore, for each level cleared the player is rewarded between 200 and 1000, in increments of 100. If the player is moved to the next level due to dying after spending too much time on a level they are awarded 0 points.

3.1.3 Capturing data with the E4 Empatica

Our method for capturing physiological signals from participants was by using the sensors of an Empatica E4 wristband [24], which can be seen in Figure 3.2, from the E4 Empatica website [3]. This device uses four different sensors to collect various information from the user;

1. The photoplethysmography sensor which measures the volume of blood in an area of tissue. This is used to calculate the blood volume pulse (BVP).
2. The electrodermal activity (EDA) sensor that measures the electrical conductivity of the skin.
3. The 3-axis accelerometer which measures movement and activity.
4. An optical thermometer that measures temperatures.

Of these sensors the only one that is not graded for clinical use is the thermometer. From these readings the E4 compiles data on several categories in the csv file format. The compressed data can be downloaded from E4 Connect, which is Empathica's web portal, and contain the following files;

[3]https://www.empatica.com/en-int/research/e4

- **ACC.csv** contains the data collected from the 3-axis accelerometer sensor in the range [-2g, 2g] sampled at 32 Hz. The accelerometer measures the movement of the wearer.
- **EDA.csv** holds the data collected by the EDA sensor sampled at 4 Hz. EDA measures the electrical conductivity of the skin and measurements have been proven to be correlated with emotions since the late 1800s [20]. EDA is also sometimes called psychogalvanic reflex or skin conductance.
- **BVP.csv** contains the data collected from the photoplethysmograph sensor, which measures the BVP, and is sampled at 64 Hz.
- **IBI.csv** stores the interbeat intervals (IBI). The IBI measures the time interval between individual heartbeats and can be used to estimate the instantaneous heart rate as well as heart rate variability. The wristband calculates the values contained within this file based on the BVP signals.
- **HR.csv** contains the average heart rate values, computed in spans of 10 seconds. The heart rate measures the number of times a person's heartbeats per minute. Similar to the IBI, these values are calculated based on the BVP signal.
- **TEMP.csv** holds the information collected by the thermometer, which is the temperature of the person playing the game expressed in degrees Celsius (°C) sampled at 4Hz.
- **info.txt** gives a brief description of all variables collected by the wristband.

Timestamps taken from UNIX time are also included in the files which makes it possible to synchronize the data with the frames collected from the game sessions.

Capturing data through the E4 gives us several types of physiological data that can be used for further analysis and experiments. Particularly BVP and EDA signals are held to be reliable indicators of emotional states in humans [36, 53]. EDA measures the electrical conductivity of the skin and measurements have been proven to be correlated with emotions since the late 1800s [20] We therefore believe that these signals will be of particular interest and utility in training emotional agents on human physiological signals.

3.1.4 Video capture

As an additional source of data that could be analyzed for emotional content, we decided to include video of participants that captured their faces, posture and movements as they played. A python script was created for this purpose, using the Open Source Computer Vision Library [6] to capture video of participants through the built-in camera on the laptop

used to play the game. The camera used was a 1.3-MP webcam attached to a Samsung Series 9 Notebook NP900X4C, which captured video at 30 frames per second with a 640×480 resolution. Video was captured in the AVI file format and stored with UNIX timestamps for synchronization. A small script was also created that starts both the game session and video capture so they run simultaneously.

3.1.5 Other equipment

The game was played on a Samsung Series 9 Notebook NP900X4C which was also used to collect the game and video data. For a better gaming experience we also decided that it was preferable to play using a gaming controller. This was deemed to be more comfortable, as well as make the controls more intuitive, as opposed to playing on the computer keyboard. A keyboard, in this context, would have a lot of excess buttons, and so especially inexperienced players would likely have trouble keeping track of the exact buttons used for each input, increasing the amount of mistakes and erroneous inputs they have while playing the game. By using a simple gaming controller we effectively counter this issue. The controller used was a wired USB controller from retro-bit, which is modeled after the original controller for the Nintendo Entertainment System.

3.1.6 Questionnaire

To provide some further context to the data in the dataset we created a questionnaire for participants to fill out. The questionnaire is designed to gather information on sex, age, handedness, prior relevant gaming experience and self-perceived excitability. The full questionnaire can be seen in the appendix of the thesis.

3.1.7 Consent form

As we gathered some data in this study that may be perceived as sensitive, we needed to create a consent form for participants to sign, in order to be able to use and share the dataset for research purposes. In the consent form we ask the participants to consent to having their in-game actions, their physiological data, as well as videos of their face recorded, shared and published for research purposes. The full consent form can be seen in the appendix of the thesis.

3.1.8 Physical setup

The gameplay sessions were conducted in a research laboratory in Oslo. The laptop and the controller was set up in an empty break-room. Pictures of the setup can be seen in Figure 3.3 on page 46.

3.1.9 Pre-trial

To test the equipment and setup we performed a trial run of the data collection study with 3 participants. This helped us identify several minor, and a few major, issues that we corrected before the main data collection study.

One serious issue concerned the video capture process and resulted in video recordings of participants being lost for the trial run. This issue was corrected by some minor adjustments to the code so as not be a concern during the main study. When reviewing the data from the trial-run it was noticed that the EDA baseline seemed to have been insufficiently established for some participants, as tonic activity kept steadily rising at the beginning of the play session for several participants, while phasic responses were largely absent. The EDA-sensor on the E4 uses dry electrodes, and is therefore dependent on the presence of some electrolytes (sweat) between the electrodes and the skin. Dry and/or cold environments may prevent a good signal from being established quickly, which may have been the issue in our case.

One participant had stable, low EDA during the first half of playing, however, readings show a much more articulated response in the second half. This participant self-reported that they felt themselves beginning to sweat about halfway through the process, indicating that the lack of EDA responsiveness in the first half could be due to a lack of sweat building up underneath the dry electrodes. Another participant, who was self-reportedly a heavy sweater, showed a much more articulated EDA throughout the play-session. This seems to corroborate the suspicion that other participants did not have enough sweat built up for proper measurements to be recorded in the early parts of the session.

These findings suggested that a physical warm-up should be added to the study protocol, to ensure that there is sufficient connectivity for the dry electrodes on the E4. Some quick testing indicated that 5-10 minutes of light exercise prior to experiment while wearing the E4 wristband helps to establish a better signal for the electrodes.

3.1.10 Protocol

A protocol, describing what data should be collected and how, was kept and continually updated throughout the preparations for the data collection. The final protocol is a part of the dataset and has also been added to the appendix of the thesis.

3.2 Performing the Study

This section details the execution of the data collection study planned in the previous section. We will discuss the participants, the data collection process itself, as well as post-processing of the data.

ID	Age	Sex	Dominant hand	Hours per week	Years active	Prior experience	Game score
0	26	Male	Right	4-8	22	Lots	17,100
1	48	Male	Left	0-1	1	Little	3,000
2	28	Male	Right	0-1	0	None	300
3	32	Male	Right	4-8	4	Some	13,300
4	32	Female	Right	0-1	5	Some	6,400
5	30	Female	Right	0-1	5	Little	2,700
6	35	Male	Left	1-4	30	Lots	14,300
7	34	Female	Right	1-4	14	Some	3,800
8	31	Female	Right	0-1	2	Little	200
9	27	Female	Right	0-1	5	Little	10,600

Table 3.2: This table shows an overview of all participants included in the dataset.

3.2.1 Participants

The data collection study was performed with 10 participants. The participants were chosen based on a small set of criteria. Firstly, we wanted to gather data from people with a broad range of prior experience in playing games like SMB, and with video games in general. The participants therefore range between those who have been playing actively for years, to people who have never touched a controller in their lives. Secondly, we wanted an even number of male and female participants. The final dataset includes data on 5 male and 5 female participants, also ranging in experience within gender groups.

The demographic information, as well as the gaming experience, of participants were gathered through the questionnaire. An overview of the participants and their answers can be seen in Table 3.2. As our pool of participants is quite small, the demographic information gathered only provides a bit of additional context to the data collections, and the data should not be viewed as representative of any particular group. However, as the tools for extending the dataset are made readily available, the potential exists for the dataset to increase the number of participants to a point where such demographic data becomes more informative.

3.2.2 Data Collection Session

Here we will give a detailed description of the process of performing a data collection session. First, a date and time was scheduled for each participant's session. After a participant arrived, and after introductions were made and so on, the participants were informed of the general outline of the data collection process. Participants were also told that there would be a competition aspect to the game sessions, and that their performance would be compared to other players. The thought behind introducing this competitive element to sessions, was to make the players feel like there was a negative consequence to dying or performing bad in the game and so be more motivated to do well, as well as be more invested emotionally. The session then progressed through the following steps;

Figure 3.3: The setting in which participants played *Super Mario Bros* for the data collection.

1. The E4 Empatica wristband is attached to the wrist of the non-dominant hand, making sure that the sensors are properly placed, and that the fit of the wristband is snug, but not too tight.

2. Participants read and sign the consent form.

3. Participants fill in the questionnaire.

4. Participants perform a 5-10 minute warm-up session to ensure that there is some sweat for the EDA sensors to connect. The warm-up consists of walking down and up 5 flights of stairs 2-3 times.

5. The participants are asked to sit alone and relax for 15 minutes in order to establish a baseline for the EDA sensor.

6. We explain the task to the participant, the goal of the game, the scoring system, the limits of the setup and introduce them to the controls.

7. The game is started, as well as the video capture, and data is collected. The participants play for 35 minutes in the described setup. A researcher observes them as they play, and may offer help at certain points, such as boss battles or when facing other special game mechanics.

8. When the session is completed, the video capture stops automatically. The E4 wristband is removed and turned off.

3.2.3 Post-processing

The video and game data was automatically stored in order on the computer. The corresponding data from the E4 was downloaded from the E4 connect web-portal and added to the set. Answers to the questionnaire and game scores were also stored in a digital format. After all data was gathered, the E4 and video data sets were cropped and synchronized with the game data based on the UNIX timestamps. Data was then sorted by participant.

3.3 Additional Synchronizing of the Data

After the data had been collected, it was discovered that there was a previously undiscovered problem with the data collection process, which had, unfortunately, resulted in certain issues with synchronizing the game frames to the video frames and sensor data. Due to the way the game loop is set up, it takes the emulator a certain amount of time to reset the environment every time the game shifts to a new level. As has been detailed, this occurs whenever the player completes, or is timed out of, a level. As the game is recording no new actions or frames during these gaps, a small amount of time is 'lost', relative to the other data collections, every time this occurs.

The total amount of game frames recorded should be 126 000, recording for 35 minutes at 60 frames per second, but is instead somewhat lower due to this issue. This means that the planned method of synchronization; simply associating each datapoint with the corresponding datapoint in the video and sensor data, will lead to an increasing misalignment of samples as the data is traversed sequentially. In the worst case there are only 124 460 game frames, representing a discrepancy of 1540 frames. In this case the real times could be distorted by almost 26 seconds towards the end of the samples.

As some players finish more levels than others the number of gaps vary between each session. Furthermore, the number of frames 'missing' per gap varies between sessions, indicating that there are no reliable, exact length that can be assumed of gaps in general or even for specific transitions.By recreating the data collection process, and measuring the length of the gaps created, it was determined that the gaps were in fact inconsistent, both across and between sessions. This means that the exact length that should be attributed each gap in each session cannot be determined, as we cannot recreate the original conditions in which they were created to gain an exact measurement. It does seem, however, that the length of gaps within a specific session does stay somewhat consistent.

With this in mind, in order to use the dataset for the types of experiments we wanted to do, we had to get a bit creative in our synchronization of data, in order to salvage as much as possible of the utility and reliability of the dataset. The lack of exact synchronization obviously somewhat reduces the usability of the dataset, however, we

are hopeful that, due to the nature of the data collected combined with the help of a tailored synchronization approach, a large majority of the synchronized data will retain its integrity, even with small temporal distortions.

For the synchronization scheme, we lean on the assumption that gaps within the same session are similar in length. Our first approach to this was based on the idea of skipping an average number of samples in the corresponding data, at the exact time where the gaps occur in the game frames, so as to ensure that each game frame corresponds to an approximately correct sample.

This method has the advantage of only using 'real' samples and thereby not introducing any potentially confusing, artificial data into the mix. The samples are synchronized relatively well in terms of one-to-one correspondence. This approach, however, does introduce certain distortions in the continuity and temporal quality of the non-game data. This happens because we are essentially losing small pockets of time, creating unnatural skips in the data and thus facilitating some data loss.

Due to the temporal nature of some of the sample sets this might make certain types of analysis, where the unbroken data pattern is important, difficult, and also might require further preprocessing of the data before this type of synchronization makes sense. For instance, to use the BVP signal, as detailed in later chapters, measurements from the E4 must be normalized across the entire recording before it is sensible to label one frame with one value. The raw data of many of our recorded signals have a temporal quality and thus would require similar preprocessing. Furthermore, as there is introduced unnatural skips in the data patterns, using the temporal information of these patterns becomes a significant challenge.

Say, for instance, that instead of one-to-one correspondence and labeling one wanted to use larger chunks of continuous data, where the exact changes in the pattern over time is significant to the result. If the data was synchronized in this way, one would have to be very careful that these sequences not contain gaps of lost data, as these would likely significantly distort the results. One would need to apply a custom data selection method, as well as use prior knowledge of where gaps occur and their session-specific length, to solve this, and would generally be significantly limited in working with the data.

Due to the issues with this scheme a new synchronization method was developed, based on the idea of padding the game data with 'replacement' frames at the points where gaps occur, so that the total number of game frames is a true representation of 60 frames per second. This second approach has the drawback of injecting artificial data and connections into the eventual sets, which might have some unpredictable consequences. This method, however, preserves the continuous, temporal nature of the data, at 60 frames per second, while still allowing for one to one synchronization of samples. This also eliminates the need for the type of preprocessing and special considerations when analyzing the data, that were required under the previous scheme, making the data much easier

to use in many contexts. These advantages seem to make this method preferable as a generalized synchronization approach.

The synchronization process for a session is thus tailored as follows. First the session is replayed and the game frames are extracted. During this process, information about the placement of gaps within the set of game frames are recorded and saved. By dividing the total number of missing frames for a session by the number of gaps, we get the average size of the gaps in that session, which tells us how many replacement frames we need to inject in each instance. The choice of replacement frames could have been one of several options, like the last frame recorded before transitioning to the new level, or the starting frame of the next level. Ultimately we chose to go with completely black frames as replacement frames, as this seemed like a good way to avoid interference between the real and the artificial data. As the game data has now been expanded to represent 60 frames per second, the other data like video frames and physiological signals can be easily synchronized based on timestamps and the known sample frequency. There is still the risk of some temporal distortions, however, the general integrity of the correlation between sample sets is largely maintained using this method.

3.4 The Toadstool Dataset

The sorted and synchronized data was gathered into a dataset which was given the name Toadstool [70]. In this section we will go through the structure and content of this dataset, as well as discuss some of its possible areas of application.

3.4.1 Dataset contents

For each participant, the dataset includes; the game actions performed at each frame, front facing video of them playing the game, the sensor data collected from the E4 Empatica in both raw and synchronized versions, and the questionnaire data. As mentioned earlier only the actions from the game sessions are stored and not the game frames themselves. There are however included several scripts for extracting the game frames and to make the dataset more accessible. The game data is stored in JSON format, the video files are in the AVI format and the E4 data is stored as CSV files.

- **participants** is the directory that contains the information of each participant. This includes the video of them playing, the controller input of each game frame, and the Empatica E4 wristband sensor data.

- **scripts** is a directory that holds a set of Python scripts meant to aid the user in getting an easy start to using the dataset. The files include a script for replaying gameplay using the provided controller inputs, a script for matching the gameplay session to the facial

expression video, and a script for matching the raw signal outputs to the gameplay session.

- **protocol.pdf** is the protocol used to collect the video game session data.
- **questionnaire.pdf** is the questionnaire that was filled out by each participant before starting the game session.
- **questionnaire_answers.csv** is a summary of all the answers to the questionnaire.
- **consent.pdf** is the consent form that was signed by each participant.
- **README.txt** is a short information file which describes the contents of the dataset.
- **LICENSE** is the file that signifies which license in which the dataset is distributed under.

Contained within the *participants* directory is a separate directory for each participant included in the dataset. Each directory has a name corresponding to the participant's ID, i.e., *participant_<ID>*, where *<ID>* is replaced with the ID of the participant.

3.4.2 Possible applications of dataset

One way to apply this dataset is for experiments similar to what we do in the later chapters of this thesis. As stated in earlier sections, our intention is to apply this dataset to the training of emotional intelligent machines using reinforcement learning. Our general approach is to train a neural network to reproduce the signals recorded from humans, on the basis of the current game state. The reproduced signals can then be incorporated into the reward function of a reinforcement learning agent and, given a strong emotional correlation, could be used to mimic human emotional responses, creating machines with a certain kind of emotional intelligence. The idea is that this kind of emotional intelligent machine might aid the pure logic of reinforcement learning models and produce improved learning, as well as possibly reveal new insights into how both machines and humans learn. They might also be a step towards machines with even more complex emotional intelligence, like the ability to recognize, express, and respond to human emotions. The Toadstool dataset, all though limited to the SMB context, can be a great starting point for certain avenues of such work. We hope that the many channels of information provided in the Toadstool dataset can facilitate a range of different, and ideally complementary, approaches to this problem, contributing to the future of this field.

Furthermore, we believe there are many other types of scenarios and research, for which the dataset could be useful. One possibility is to use it to detect relationships between the player sentiment and the current gameplay state. This could be done solely based on the gameplay frames

and collected sensor data, or could also be combined with the player's facial expression for further analysis. The uncovered relationships could be interesting when studying how players get invested in video games, and what typical scenarios contribute to a strong reaction from players.

Another example could be predicting a person's facial expression based on a given game state or degree of progress in a game or level. This could also be expanded to the sensor data, as one could predict the sensory output of the wristband using the game state and facial expressions as input, like photoplethysmography [56] does with the heart rate, but connected to the current game state the player is in. These two problems could be modeled as either a regression or classification problem, depending on the application.

The dataset could also be useful in the field of game design. Understanding the correlations between game progress, game state, the input of players, and the emotions and sentiment of the players could potentially enable a new approach to experience-based game design. With the possibility to use this as a method for playtesting, game designers can evaluate when and how their crafted experience is (or is not) invoked in the players. Moreover, games can be built around the concept of self-adapting challenge and difficulty by counter-acting unnecessary frustration, boredom, or annoyance by adapting the game to the player's emotion and sentiment. Last but not least, the correlation between the game, emotions, sentiment, and game engagement, especially flow (the state of being fully immersed in an activity while enjoying it), can be investigated [39, 71]. This could lead to a better understanding of what makes games enjoyable and impact fields like game and media studies, psychology, game engineering, game design, and educational games.

Chapter 4

Simulating Emotion using the Blood Volume Pulse

After having created the necessary dataset, the next step was to use the data collected, and find a way to train an algorithm to predict a signal that correlates to the recorded emotional reactions. As we have seen in the previous chapters, the Toadstool dataset [70] provides a variety of data that can be used, independently or in combination, to attempt to achieve this. As our project must maintain a limited scope, we do not have the opportunity to investigate the many different possible implementations here. This is, however, an interesting available path for future work.

Instead, we rely upon what has been shown to be effective in the past. Once again, we draw inspiration from the work in the *Visceral Machines* project [46]. As discussed in section 2.7.4, this work used a deep neural network to predict the BVP in a self-driving system. Our approach will be to apply the same principle in our own context of playing Mario, following the same process as closely as we reasonably can. In this chapter we will briefly explore the methods and history of the BVP measurement, as well as describe in detail the procedure we used to obtain, process, and train neural network models to predict, the BVP signal amplitudes. We also perform some evaluations of the performance and behaviour of the trained models.

4.1 Photoplethysmography, Fight or Flight, and Leveraging the Blood Volume Pulse

In this section we will look at the measurement of photoplethysmography (PPG) and how we can use it to obtain the BVP. We will also discuss how the BVP is related to visceral emotional reactions in the sympathetic nervous system, and how these can be leveraged for reinforcement learning.

4.1.1 Photoplethysmography

Since the first discovery of the oximeter during World War II, as a way to estimate hemoglobin saturation and assess the oxygenation of pilots,

the technique known as pulse oximetry has seen great improvements in technology, and has established itself as a major tool in clinical medicine. Early oximetry devices, that collected measurements via the earlobe, where cumbersome to use, requiring much skill and calibration, and could even burn patients, because of their reliance on heating the earlobe [72].

In contemporary times however, pulse oximetry measurements can be obtained both easily and non-invasively through sensors on the fingertips or wrist. The technology works on the observation that certain substances, like hemoglobin, absorbs particular light wavelengths in a selective way. By reflecting light off of tissue and analysing the light reflected back, an estimate of the saturation of certain substances can be obtained.

Modern photoplethysmography sensors, like that on the E4, take advantage of this principle to obtain optical plethysmograms that can be used to determine the changes in blood volume of a tissue. This is what is called the BVP, and is essentially a measurement of the amount of blood rushing through an area with every beat of the heart. Potential measurement sites are anywhere where a pulse can be accessed. Popular measurement sites for BVP include the pads of the fingers or earlobes. In our case we use the wrist as a measurement site, through the E4 wristband.

4.1.2 The E4 PPG sensor

The E4 obtains the BVP signal by use of a proprietary algorithm and a PPG sensor that combines both red and green light exposure. It has a sampling rate of 64Hz. The sensor measures variations in blood volume in arteries and capillaries that indicate changes in the blood flow. PPG sensors typically accomplish this by emitting an infrared light onto the surface of the skin, and reflecting it back to the PPG sensor. Because red light is selectively absorbed by the hemoglobin in red blood cells, the light returning to the sensor will indicate the relative volume of blood in the tissues. With the E4, the inter-beat interval (IBI), the time interval between heartbeats, is also calculated using the BVP measurement. The immediate heart rate is also derived from the BVP.

4.1.3 BVP and the Sympathetic Nervous System

The sympathetic nervous system is a substructure of the autonomic nervous system. The autonomic nervous system is also known as the involuntary nervous system, as it regulates bodily functions such as heart rate, body temperature, sweating, blood pressure and digestion, without any conscious direction [45].

The sympathetic nervous system is responsible for the body's involuntary reaction to dangerous or stressful events. When such an event occurs this system releases a flood of hormones that raises the alertness of the body, increases heart rate and pushes extra blood to the muscles, preparing for an immediate reaction to danger. This viscerally emotional effect is what is often referred to as the "fight or flight" response.

The effect is so quick that it can be difficult to realize that it has even occurred, thus enabling people to react to danger before being fully conscious of what is happening. For instance, someone may jump out of the way of a falling object, before being consciously aware that it is falling towards them.vIn addition to protecting us from danger, the physiological changes play a role in our appraisal of emotion and our decision making. Even though this system is prone to mistakes, especially false positives, it is vital for human beings as it enables us to make quick decisions in stressful situations.

BVP is a reliable indicator for changes in the sympathetic nervous systems, as the PPG sensor is directly based on detecting vasomodulation. Vasomudulation refers to the changes in the amount of blood running through a tissue area with each pulsation of the heart. When the "fight or flight" response is activated, the body diverts blood from the extremities to the organs and muscles to prepare for action. When this happens the waveform of the BVP pinches due to the increased vasoconstriction. This principle has been widely applied in *Affective Computing* scenarios, for instance for capturing emotional responses in marketing and media testing [82], and for assessing users frustration when playing a deliberately slow video game [62].

4.1.4 Leveraging the BVP for Reinforcement Learning

In the *Visceral Machines* paper [46], McDuff and Kapoor proposes a possible connection between the sympathetic nervous system response and dangerous situations for the driving agent, which their experiments provide support for. This means that introducing this response into the learning system can provide the agent with useful situations about what situations to avoid. In the case of playing Mario, such a response might be similarly connected to situations where the player is close to dying on a course.

To assess the sympahtetic nervous system response of their subjects, McDuff and Kapoor rely on measurements of the BVP. As we have seen, the BVP is a good indicator of the sympathetic nervous system's responses. By collecting the real BVP data from humans, and training a deep neural network to simulate the signal based on the visual information in the environment, we are able to introduce an intrinsic, emotional reward system into the agent, which will hopefully complement the extrinsic reward represented by the agent's logical understanding of the environment.

4.2 Processing the BVP data

Before the BVP data collected from the E4 could be used for training a deep neural network, certain pre-processing of the data was necessary. In this section we will go through the process used to transform the data into a form that is suitable for our purposes.

4.2.1 The BVP signal

The BVP, as given by the E4, represents a waveform. The details of the BVP signal can be seen in Figure 4.1, from the E4 Support website.[1] The high point of the wave is called the systolic peak, while the lowest point is called the diastolic peak. The distance between diastolic peaks can be used to calculate the heart rate and inter beat interval (IBI).

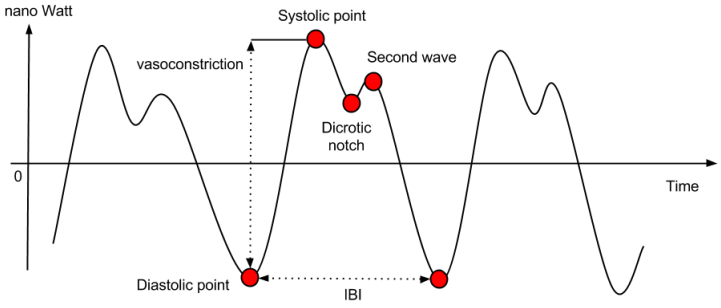

Figure 4.1: Model of the BVP signal from the E4 Empatica.

The difference between the diastolic peak and the systolic peaks, i.e., the amplitude of the signal, represents the subjects vasoconstriction. This is the measurement we are interested in as it is the best immediate indication of a sympathetic nervous response, as seen in the previous chapter. The range of the actual nano Watt values differs from person to person, due to physiological differences, but is stable in relation to itself across a recording. The diastolic and systolic points are proportional to each other.

4.2.2 Normalizing the Recording from Waveform to Amplitude

As we are interested in the amplitude of the signal, as an estimation of vasoconstriction, some processing of the original BVP signal is necessary to produce the set of samples we want to set as the ground truth for the frames we train our deep neural network on.

In our project the aim is to associate the frames from the game with the estimated vasoconstriction of the person playing, at that time-step. Simply feeding a network the game images with the corresponding values from the BVP as is, would not make much sense, as this point could be any part of the waveform, and would not give any reliable indication of the amplitude or vasoconstriction of player. In this section we will walk through the steps we used to transform the original BVP signal, into a sample set of associated amplitudes that could be used for training our CNN models. The different stages of transformation is illustrated in Figure 4.2

The first step is to normalize the signal. As the range of values varies widely from person to person, and are not conducive to our machine

[1] https://support.empatica.com/hc/en-us/articles/360029719792-E4-data-BVP-expected-signal

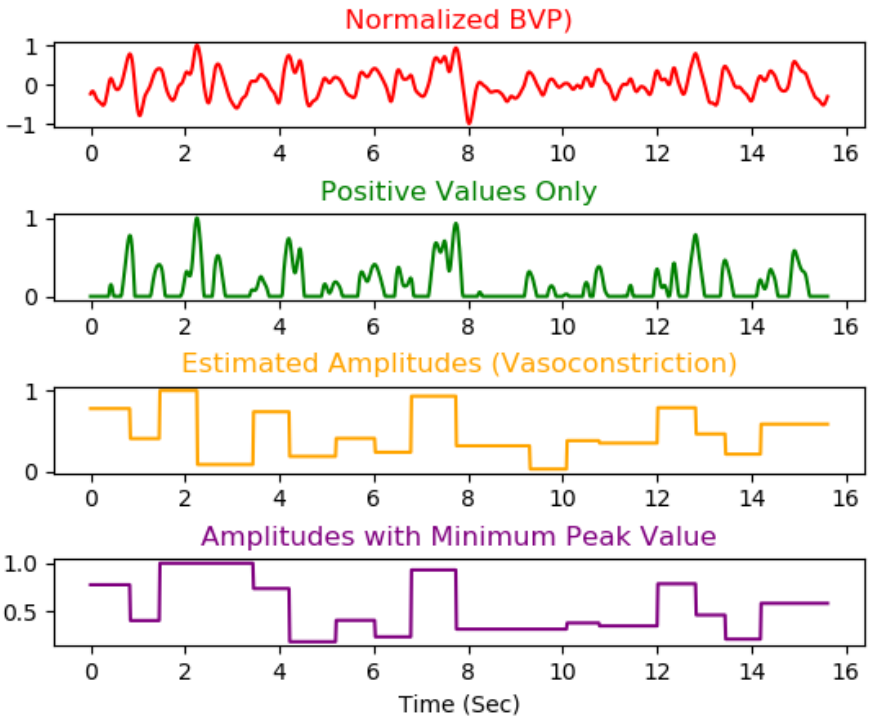

Figure 4.2: The stages of transforming the BVP signal.

learning algorithm, the range of the signal is normalized into a range between -1 and 1.

Following the method used by McDuff and Kapoor [46] we will use only the systolic peak values, relative to 0, to estimate the vasoconstriction. The signal is therefore pruned to only include positive values. Negative values are replaced with 0.

Next, we estimate the amplitude associated with each sample index, creating a new set consisting of the list amplitudes that corresponds best to the given time-step. To locate the systolic peaks in the BVP we used the *find peaks* method from the *signal* package, which is a part of the ScyPy library [75].

The minimum distance between peaks was set at 40, meaning that detected peaks closer than this will be skipped. This is necessary as the signal has minor peaks that appear between heartbeats, which are mostly not indicative of the actual vasoconstriction.

The peak values found are duplicated across the index space between the peak current peak and the last one, creating an even stream of estimated BVP amplitudes, based on the systolic peak values.

Additionally, due to some instability in the signal, values that fall below a certain threshold, and are likely to be inaccurate measurements, are replaced by their predecessor value. The minimum peak value is set at 0.1 for the normalized recording.

The final step is to modify the rate of samples in the set, so as to best accommodate the rate of the game frames. The BVP samples are given at 64 Hz, while the game frames are collected at 30 frames per second. Therefore, to achieve a 1:1 ratio between the two sets of data, we remove every 16th sample from the BVP amplitudes, before halving the sample set by removing every other sample. The remaining BVP amplitude samples can then be conveniently matched to a single game frame each.

4.3 Predicting the BVP amplitude

The estimated BVP amplitudes were used to train a deep neural network to predict an amplitude based on a frame from the game. In this chapter we will detail the method we applied, as well as the model architecture and the results of the training.

4.3.1 The Neural Network

The setup for the neural network used to predict the BVP amplitudes was modelled on the one used by McDuff and Kapoor, in their *Visceral Machines* project [46]. This was a CNN with seven convolutional layers and one dense layer. When building our own model, we followed the specifications laid out in their paper as much as reasonably possible. The input to the model is a single image, and the ground truth is the corresponding estimated BVP amplitude. The resulting CNN is a regression type model, meaning it produces an output in a numerical range. The architecture of the CNN model can be seen in Table 4.1.

Layer no:	Type:	Parameters:
0	Input	input_shape = (84,84,1)
1	2D Convolutional	filters=32
2	2D Convolutional	filters=32
3	Max Pooling	pool_size=(2, 2)
4	2D Convolutional	filters=64
5	2D Convolutional	filters=64
6	Max Pooling	pool_size=(2, 2)
7	2D Convolutional	filters=64
8	2D Convolutional	filters=64
9	2D Convolutional	filters=64
10	Max Pooling	pool_size=(2, 2)
11	Flatten	-
12	Dense	units=128
13	Dropout	dropout_rate=0.5
14	Dense (Output)	units=1

Table 4.1: The architecture used for the CNN models trained to predict the BVP amplitude. All convolutional layers uses a kernel size and strides equal to 3.

All the convolutional layers of model, as well as the first dense layer, uses the rectified *linear unit activation function*, or *ReLu*, while the last layer uses the *linear* activation function, to create the desired form of output. The model was compiled using the *Adam* optimizer with a learning rate of 0.0001, and a *mean squared error* loss function. Except for the exact kernel sizes and strides, for which we had to make assumptions, this model has the same characteristics as the one used by McDuff and Kapoor [46].

4.3.2 Preparing the data

Each sample image from the game session was given a discrete identifier, associated with the corresponding BVP amplitude. The data was split into training and validation set, with approximately 25% going towards validation and 75% going towards training. The nature of the split was done on a second by second basis, to capture a broad range of encountered situations in both sets, with every forth second being used for validation. After the split, the samples in both sets were randomly shuffled.

As the set of all the images used for training was quite large, it was impractical to load the entire dataset into memory during training. Therefore a custom data loader structure was built to feed batches of data to the neural network when needed. The images were sampled down to a quality of 84×84 pixels and grayscaled. They were then loaded into numpy arrays and normalized, by dividing every pixel value with the max pixel value (255), before being fed to the training algorithm.

4.3.3 Training and Results

We trained one CNN per participant. The reason for training models on a single participant is that we want the model to approximate the emotional response of a distinct person, and not be a mix of different, possibly interfering responses, from many people at once. Mixing, aggregating, prioritizing etc. of the emotional responses of multiple people is however, an interesting avenue for further work.

The models were trained with a callback method that monitors the mean absolute error on the validation set, and stops the training after ten consecutive epochs with no improvement in this metric. The model from the best performing epoch was saved to be used for the reinforcement learning experiments. The models trained in a range of 11 to 49 epochs.

For a simple evaluation of the CNN models, we also calculated the error when using the mean of the response variable in the training for prediction. This measurement is also called ZeroR. Comparing the ZeroR with our results should give us some indication about how well our model performs at predicting the BVP amplitudes. The performances from the best training epoch for each model and the corresponding ZeroR can be seen in Table 4.2.

From these experiments we can see that most of the trained models, eight out of ten, slightly outperform the ZeroR on the mean absolute error metric, while a smaller minority, two out of 10, outperform the ZeroR on the root mean squared error. All tough a more significant improvement

	CNN		ZeroR	
ID	MAE	RMSE	MAE	RMSE
0	0.090	0.116	0.075	0.099
1	0.098	0.135	0.103	0.131
2	0.073	0.109	0.075	0.100
3	0.037	0.059	0.050	0.070
4	0.052	0.084	0.069	0.094
5	0.090	0.128	0.094	0.121
6	0.088	0.124	0.090	0.116
7	0.104	0.146	0.109	0.139
8	0.064	0.103	0.060	0.090
9	0.099	0.135	0.109	0.134

Table 4.2: This table shows the results the experiments of trying to predict the BVP amplitude using video game frames.

would have been desirable, due to restrictions on time these models will have to suffice as providers of the emotional signal for our experiments. All though the models can not be assumed to reliably reproduce a truly plausible BVP amplitude, at least can be said to produce a non-random signal based on human emotional arousal data.

To gain a better understanding of what these models actually produce, we collected one thousand test images from different levels of the play sessions in the dataset, and had the CNN models predict values based on this common set of frames. Table 4.3 shows the ranges of values, as well as the mean values, predicted for each model.

We can observe that some of the models show large variation here, some larger ranges of values, up to variations of about 0.9 at the largest, while several others only produce predictions in a much narrower range. Furthermore, we can see that all the models produce somewhat similar mean values, in the range of 0.149 to 0.289. One likely reason for the narrow ranges could be that these CNN models are converging more around a mean value. Other contributing factors could be the existence of outlier peak values in the BVP data, which may results in a large majority of the amplitude values used for training being in a narrower range than the true range of the set, for some of the models.

To begin to analyse he incentives produced by the models, i.e., what types of game situations would create what kinds of values, we manually reviewed the predictions given to the lower and higher scoring groups of test frames for the model trained on participant 0. From this cursory look we could see that much of the predictive variation could still be determined by level, or world, specific identifiers, like the differences in backgrounds among levels. For instance, the top 2%, meaning those with the highest associated prediction values, of the test images, and likely much of the top range, where mostly dominated by frames from levels using a dark background. These are either castle levels, or certain other late levels

Values predicted by CNN:				
CNN ID:	Low:	High:	Range size:	Mean
0	0.226	0.374	0.148	0.281
1	0.127	0.653	0.526	0.221
2	0.14	0.371	0.231	0.194
3	0.087	0.565	0.478	0.149
4	0.108	0.556	0.448	0.196
5	0.065	0.968	0.903	0.229
6	0.14	0.442	0.302	0.24
7	0.095	0.925	0.83	0.289
8	0.119	0.512	0.393	0.195
9	0.104	0.702	0.598	0.266

Table 4.3: This table shows the lowest, highest, range size and mean values predicted by the trained CNN models, based on 1000 test images collected from the play sessions. The CNN ID is the same as the ID of the participant used to train the model.

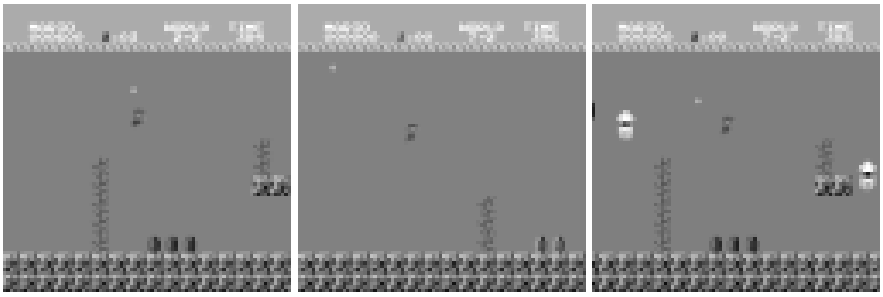

Figure 4.3: Some examples of images with predicted values in the bottom 2% of the test image set (1000 frames), using the CNN based on participant 0.

that use different background graphics from the rest of the levels. Some examples of such frames can be seen in Figure 4.4.

On the opposite end of the spectrum, frames taken from water levels, which offer their own distinct set of game mechanics, populated most of the bottom 2%. This might indicate that the nature of the arousal response changes throughout the game session or are associated with specific levels or features of longer stretches of game play. We would have liked to see if the models show a reaction to specific situations, however this limited investigation revealed no such connections. Whether there indeed exists relative differences in the predictions based on more specific situations remains an open question based on this analysis.

Ideally we would have wished to perform some more extensive experiments to test the behavioral incentives produced by the CNN models at this point, before moving on to implementing them into a reinforcement learning framework. For instance, by collecting sets of test frames that

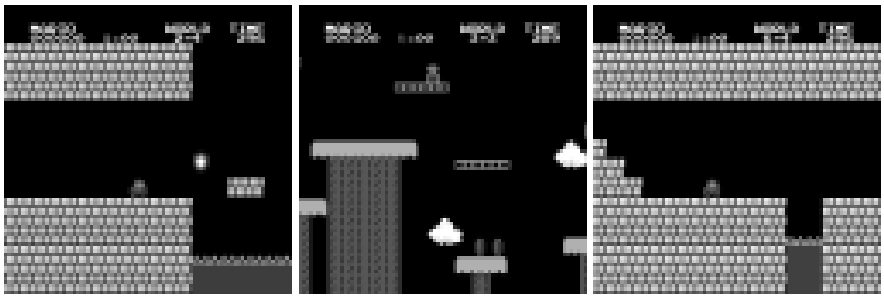

Figure 4.4: Some examples of images with predicted values in the top 2% of the test image set (1000 frames), using the CNN based on participant 0.

represent distinct game situations, one could begin to measure the typical values predicted for these, and thereby gain a better understanding of what situations are likely to produce what reward for what model etc.. However, as we must maintain a narrower scope for this thesis, we limit ourselves to this reduced, and somewhat superficial, investigation of the behaviour of this single model. This approach might be limited in terms of offering actionable insights, but may still serve as a rudimentary example of a way to begin such an investigative process.

Chapter 5

Building a Double Deep Q-learning Network to play Super Mario Bros

In order to explore how introducing an emotional component into a reinforcement learning agent would affect the learning process, we first needed a more standard reinforcement learning learning algorithm,, that performs decently in the SMB environment, and which could serve as a baseline for comparisons. The model we built for this purpose was a type of Q-Learning network, or more specifically a Double Deep Q-learning Network (DQN). As detailed in section 2.4.3 the DQN is a model-free reinforcement learning algorithm, implementing the experience replay method, and having two separate deep neural networks, one for choosing the next action and one for predicting future values.

5.1 Environment and Wrapper Functions

The environment that the DQN agent is trained in is the SMB game, as given by the gym-super-mario-bros [37] reinforcement learning framework, which is the same one used for collecting the dataset. Due to limitations on time, as well as other resources, the model was not trained on the full version of the game, but was constrained to learning on a single course, in order to reduce the scope of the training experiments. Several wrapper functions were also applied to the environment to further customize it for efficient learning. We look at some of these in this section.

5.1.1 Limiting action space

The gym-super-mario-bros environment by default uses the complete action space of the original game, in which every combination of button presses in the game are represented. This produces a space of 256 discrete actions, most of which are never used if playing seriously, introducing an unnecessary level of complexity to the problem the learning algorithm tries to solve. For this reason the gym-super-mario-bros framework provides

the JoypadSpace wrapper, which limits the set of legal actions to a more reasonable selection. The JoypadSpace wrapper gives three options for legal action sets;

1. Right only; this is the simplest move set. It only allows for basic actions and no left moving actions.

2. Simple movement; only basic actions but moving to the left is allowed

3. Complex movement; allows for more complex actions like ducking

We chose to use the 'simple movement' set of actions, to achieve a nice balance between simplicity and range of behaviour.

5.1.2 Skipping frames

Game frames produced by the environment, that are immediately adjacent to one another, are very similar to each other, to the point of being virtually identical. This means that if we were to capture each frame as a memory in the replay buffer, we would end up with an unnecessary number of memories that were virtual duplicates of each other, which would not benefit the training process. To address this a wrapper function was applied that produces only a single game frame for every four frames produced by the environment, effectively skipping three out of four frames. The function picks one of the middle frames and returns it with the total accumulated reward for all four frames.

5.1.3 Stacking frames

In order to incorporate a temporal dimension, and a sense of direction, in the data, a wrapper was added that stacks the four last frames on top of each other. This counteracts the static nature of the still images from the game and introduces information about the immediate past of the frame, giving the model the opportunity to differentiate between similar looking situations.

5.1.4 Warping the frames

The game frames are given by the environment in 240×256 pixels in color. Images of this size take quite a lot of time to process by the model and may contain a lot of superfluous information. In order to speed up the training training process a wrapper layer was added that scales down the quality of the images, before they can be stored as memories to train the algorithm. The images are also grayscaled, which turns the image into black and white. This drastically reduces the size of the images as the system then only needs to store one dimension for each pixel, representing the scale from white to black, instead of three separate ones, for red, green and blue values.

5.2 Reward

The reward function provided by the gym-super-mario-bros framework is designed to make Mario move as far right as possible, towards the end of the level, as fast as possible, while avoiding death. To achieve this, three separate variables are used to shape the reward;

1. The difference between states of Mario's location on the level. If Mario has moved to the right he receives a positive reward, if he has moved left he receives a negative reward, and if he stands still he receives no reward at all. The reward is equal to the change in the games x-coordinates.

2. A time penalty based on the game clock. This prevents the agent from standing still. The negative reward is equal to the change in the game clock.

3. A death penalty that gives a larger negative reward (-15) if the agent dies in a state.

The variables are summed together and the reward is clipped into a range of between -15 and 15. In addition, we apply several wrapper functions to the environment which further shape the reward received by the algorithm.

5.3 The Neural Networks

The design used for the neural networks part of the DDQN was a simple CNN based upon models used in similar projects. An overview of the architecure of the CNN can be seen in Table 5.1. The CNN is implemented using keras' sequential model, with all the layers using the rectified linear unit activation function (ReLU). The model is compiled with the *Adam* optimizer and a mean squared error loss function.

Layer no:	Type:	Parameters:
0	Input	input_shape = (84,84,4)
1	2D Convolutional	filters=32, kernel_size=8,strides=4
2	2D Convolutional	filters=32, kernel_size=4,strides=2
3	2D Convolutional	filters=32, kernel_size=8,strides=4
4	Flatten	-
5	Dense	units=512
6	Dense (Output)	units=(size of action space)

Table 5.1: The architecture of the neural networks used in the DQN.

5.4 Optimizing the parameters of the DQN Agent

We ran several different experiments in order to improve the performance of the DQN agent in the environment. We created a range of versions of the agent, by adjusting certain key parameters in the system. We could then compare the performance of the different intermediate versions of the agent, and pick the implementation with the best performance going forward. The final implementation uses a learning rate of 0.0005, a memory replay buffer of size 10 000, and downscales the frames to a 84×84 pixel quality. In this section we go through the different iterations of the algorithm, and show how we arrived at the final implementation.

It is important to note that the goal of this project is not to build a highly optimized reinforcement learning agent. There are many parameters, including the ones described in the environmental setup like the amount of frames to stack, that could be adjusted with the aim to further improve the performance of the agent, possibly far beyond what is achieved here. Such a process is however a laborious and time-consuming one, and falls outside the scope of this paper. Even so, a certain amount of optimization does seem desirable for our purposes, as performing experiments on a reinforcement learning agent with very poor performance would not only be more difficult, as comparisons would be harder, but would also certainly make the results less interesting, as they would be further removed from a scenario that is truly relevant to the field.

5.4.1 Adjusting the learning rate

The learning rate, in a machine learning setting, refers to how quickly the algorithm learns from new information. It determines how much the model will update its internal weights with each training cycle, or essentially to what degree new information will override old information in the neural networks. The learning rate is a key parameter for reinforcement learning algorithms like our DQN, and can have drastic effects on the models performance.

If the learning rate is too high, meaning the agent replaces old information quickly, the learning becomes too rapid and the result is that the agent rushes to learn a sub-optimal policy and becomes unable too improve. When the DQN becomes too fixated on a policy, it keeps repeating the same patterns over and over, limiting its range of behaviour. This is exactly what we want it to do, given that the policy is actually close to optimal. When the agent becomes fixated on a sub-optimal policy however, the narrow range of behaviour becomes a roadblock for further improvement.

On the other hand, if the learning rate is too low, the agent may take a very long time to learn an optimal policy. The slow adaptation of new information results in the model needing to see a lot of similar data on a given context before significantly improving in that direction. In the worst case, learning becomes too slow and the model can get stuck, unable to digest the necessary information to improve its behaviour patterns. The

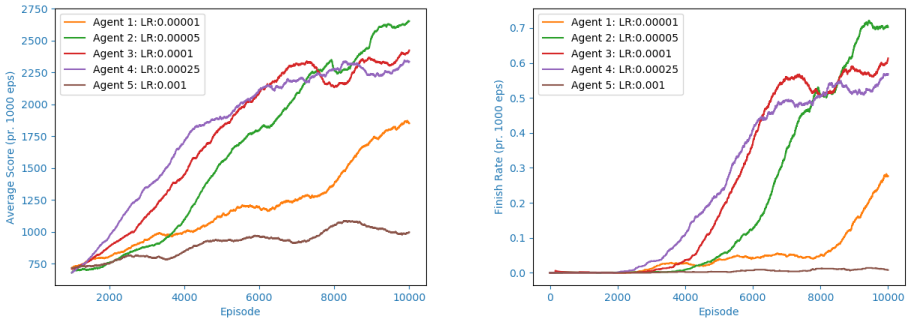

Figure 5.1: Comparing different learning rates over 10 000 episodes.

ideal learning rate, therefore, is one that is appropriately small for a given context, all the while not being so low that it significantly impedes the learning process.

We ran the DQN model using a range of different values as the learning rate. Figure 5.1 shows the performance of the algorithm, using five different implementations of the learning rate. Performance is measured in terms of the average score achieved, as well as the average finishing rate, which refers to how often the agent was able to finish the level. Averages are calculated as the mean values for the last 1000 episodes. The maximum score achievable by the agent in this scenario is a little over 3000. The average finish rate is given as a value between 0 and 1.

The poorest performer in this episode range was the implementation of agent 5, with the largest learning rate at 0.001. This version struggles to get the average score to a comparative level and barely registers in finished games. This is a clear indication that the model has prematurely converged on a sub-optimal policy. The same tendency can be observed in agent 4, which uses a learning rate of 0.00025. All tough this implementation performs significantly better than agent 5, and in fact achieves the fastest, early improvement of all the compared agents, it also converges on a sub-optimal policy, and struggles to improve beyond a certain level of performance.

When looking at performance over 10 000 episodes, agent 2 and 3 clearly do the best out of the compared models. While somewhat slower in the beginning, agent 2, with a learning rate of 0.00005, achieves the best performance towards the end of the episode range. Agent 1, with the lowest learning rate of 0.00001, is much slower to improve than agent 2 and 3, and yet it does begin to improve quickly around episode 8000. Agent 3, with a learning rate of 0.0001, is a little slower to improve in the beginning than agent 4. However, between episode 5000 and 6000, it overtakes the performance of agent 4. Even so, the implementation of agent 3 still suffers from the same issue, and converges on a sub-optimal policy before it has achieved the same levels of performance as agent 2.

This picture becomes even clearer when we look at the further progression of the training. Figure 5.2 shows the performance of the three

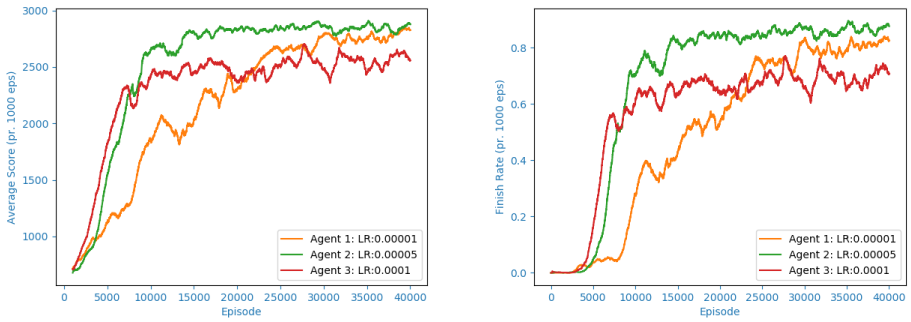

Figure 5.2: Comparing different learning rates over 40 000 episodes.

agents with the lowest learning rates, over 40 000 episodes. In this episode range, it is clear that agent 3 has become stuck on a sub-optimal policy. In comparison, agent 2 has maintained good improvement and is achieving close to optimal results around episode 15 000. An interesting observation is that agent 1 continues to improve over this range, eventually achieving a performance that is comparable to agent 2 towards the end of the 40 000 episodes. This indicates that the model is fully capable of developing knowledge and learn at this lower learning rate, however, it also shows the drawback of setting the learning rate too low, as the training process is significantly slowed down .

Based on these experiments, the most appropriate learning rate we have tested seems to be that of agent 2, at 0.0005. This implementation maintains a good balance between learning efficiency and high performance, reaching most of its potential after within 10 000 episodes. It is surely possible to further optimize the learning rate of this model, but as discussed, that is outside the scope of this thesis.

5.4.2 Adjusting the size of the models memory

As discussed in section 2.4.3, an important element of our double deep Q-learning network setup is the implementation of experience replay. This requires a dynamic memory bank, often referred to as the replay buffer, which contains the instances, or memories, that our algorithm trains on. In the case of our setup the replay buffer contains memories with the following information:

- The state the agent is in before the action
- The action taken
- The state following the action
- The reward given to the transition
- Whether the resulting state is a terminal one.

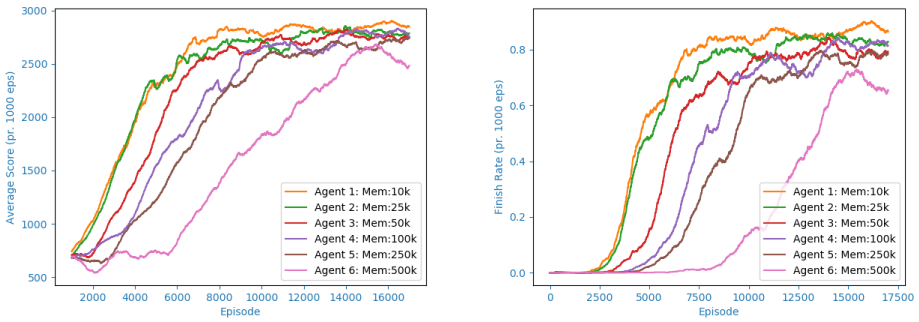

Figure 5.3: Comparing different sizes of replay buffer over 17 000 episodes.

Memories stored in the replay buffer are collected in batches with each training step of the model. As the buffer reaches its maximum capacity for memories, it will begin to overwrite the oldest information in its storage with newly collected data. This means that the predetermined size of the replay buffer can severely impact the training of the algorithm. If the size of the buffer is small, old memories will be quickly overwritten in favour of more recent experiences, making the old ones unavailable as training data. This will lead to an exaggerated focus on new memories, biasing the learning towards the path it is currently exploring. Setting the buffer to be very large can remedy this effect and preserve older information for longer, however, this might also have drawbacks, for instance due to older memories being less productive towards an optimal policy and therefore slowing down the learning process.

All though the implementation of experience replay is by now a common practice, the exact effect this has on learning is still quite poorly understood. In the case of memory, it has been shown that both too little, and too much, memory can slow down the learning process, however there is still much to be discovered before an exact determination of the ideal memory size for a given context can be determined prior to testing [42].

For our model we tested six different variations of the size of the replay buffer, using the same metrics of average score and success rate, with sizes ranging from 10 000 memories at the low end, up to 500 000 at the maximum. The results can be viewed in Figure 5.3.

This test shows a clear tendency that, for our specific context, as well as the specific range of buffer sizes tested, smaller sizes of the replay buffer performs better than higher ones. While most of the agents are able to reach comparative performance given enough episodes, a substantial difference can be seen in the amount of episodes needed to achieve their potential. The clear worst performer is agent 6, with space for 500 000 memories, while the best performer is agent 1, with the smallest memory size of 10 000. Agent 1 slightly outperforms agent 2, with buffer size 25 000, in terms of results and training speed. The correlation between smaller buffer size and performance is consistent throughout the testing results.

More fine-tuning of this parameter may be possible, as well as implementing specific training schemes that uses the replay buffer in different ways, like prioritizing certain memories over others, to make the algorithm even more optimized. It is also worth noting that other similar but different setups, like training the Mario agent to complete more than one course at a time, might gain more benefit from a larger replay buffer. However, for our purposes, and based on the results of our testing, a replay buffer with room for 10 000 memories seems to achieve an appropriate and sufficient level of performance.

5.4.3 Adjusting the quality of the images

Another important decision for the algorithm is what quality, or pixel size, the images used for training should be down-sampled to. This is not only a question of what quality of images contain more relevant information, as restrictions on computational power and memory must also be considered. Images of a higher quality will of course, retain some additional information, however, a larger image size means a lot of extra work for the CNNs, as well as other image related processes like saving and loading, which will significantly slow down training. Larger images also means that more memory is consumed, as the size of the replay buffer, in terms of RAM, will expand relative to the image sizes.

A full quality image can also be counter productive, as the additional information in these may be unrelated to the agents goals and will therefore be slowing down the process without adding anything of benefit. For many environments, a severely downgraded image can retain most, if not all of, the elements needed for the agent to learn and complete its task. In fact, for video games like SMB, much of the visual information is only there for aesthetic purposes or to increase human enjoyment, and bears no actual relevance for the progress of the game itself.

The goal then is to downgrade the quality of the images used, to a point where they still retain most, or at least sufficient amounts, of the information necessary to be used effectively in training, but also allows for a relatively efficient training process. The image size of the game frames collected from the standard gym-super-mario-bros environment are given at 240×256 RGB-colored pixels. As discussed, training on the full quality images would lead to an unnecessarily slow training process, and so the images are automatically downsized before being stored as memories. This happens in the warping wrapper discussed in section 5.1.4.

The warping function firstly grayscales the images, reducing the size of the image array by a factor of 3, as we now only need to store a single intensity value per pixel, instead of the three values required by a image in full color. Furthermore, it warps the image to a given smaller size. We tested three different agents, training on different image sizes, in order to compare their performance and efficiency; Agent 1; 60×60, Agent 2; 84×84 and Agent 3; 120×120. The results of comparing the performance and episode count, over the same training time on equivalent hardware, can be seen in Figure 5.4.

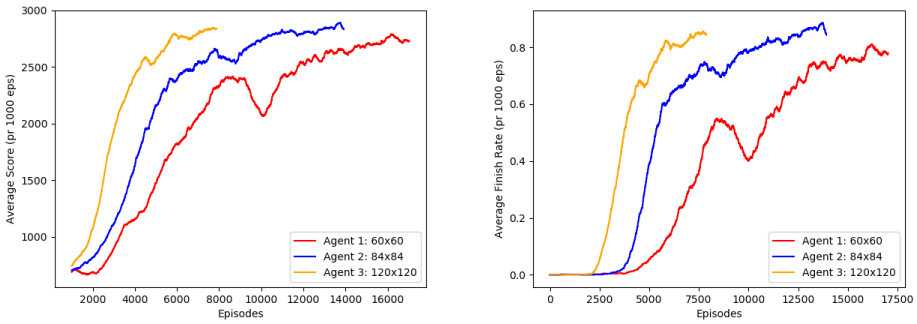

Figure 5.4: Comparison of running the DQN with different image pixel sizes. The Agents were run on equivalent hardware over the same amount of time.

The result of this experiment shows that, over the same time period. Agent 3 completed well over double the amount of episodes as Agent 1, showing the efficiency of this lower quality. However Agent 3 is not able to achieve the same performance level as the other agents within the given time frame, indicating that too much information is lost in the downsampling of the image.

Agent 1, with its higher quality images, reaches good performance levels over a lower number of episodes, and clearly shows the most effective progress per episode. However, the efficiency of the process is also limited by the large images, resulting in it taking a long time to complete episodes.

Agent 2, in comparison, trains slower per episode, but is able to complete almost twice as many episodes as Agent 1 in the same time. Agent 2 also achieves a marginally higher performance towards the end of the episode range. In the end, we chose to go with the 84×84 pixel size images, exemplified in Agent 2. Based on the observations from our testing, this seems like a reasonable candidate for our purposes, showing a decent balance between performance and efficiency, as compared to the other options.

Chapter 6

Introducing Emotion to the DQN

The final objective of this thesis is to integrate a recreated emotional signal into a reinforcement learning agent in order to explore how this might affect the agents behaviour. In chapter 3, we saw how we collected the real data from humans playing SMB. In chapter 4, we used the collected BVP data to produce models that predicted the BVP amplitude, an indicator of visceral emotional responses, of a participant based on the game frames. In chapter 5, we built a standard DQN model to play SMB that will a serve a a baseline for comparisons when introducing the emotional signal. In this chapter, we put these components together to introduce emotion into the DQN agent. We then compare how the new emotional models perform, in relation to each other and the version without emotion.

Note that the experiments performed in this thesis are of a preliminary and somewhat exploratory nature. That is to say; we aim to demonstrate a method in which CNN models that produce an emotional signal can be integrated into a reinforcement learning (RL) context, as well as observe phenomena that might be interesting for further exploration, rather than determine any definitive conclusions on the efficacy of our approach.

6.1 Theoretical framework

As has been discussed in earlier chapters, the idea for our experiments is to supplement the inherent, logical abilities of a classic reinforcement learning algorithm, with a signal that represents emotional reactions to the environment. This signal is represented by the predicted BVP amplitudes as an estimation of vasoconstriction, which is used as an indicator for arousal of the sympathetic nervous system.

From a psychological perspective, our approach to emotion is a dimensional one, as we attempt to measure a certain type of activation along the dimension of emotional arousal. In earlier chapters we discussed how the activation of the sympathetic nervous system is associated with the "fight or flight" response, which entails emotions of immediate fear, alarm and tenseness. As we have seen in section 2.3, these emotions

can be located towards the high end of the arousal dimension of Russel's circumplex model, seen in Figure 2.3, on page 17. Towards the opposite end of the same dimension, we find feelings of calm, relaxation and satisfaction.

A reasonable assumption seems to be that the range of arousal relevant to the BVP amplitude, is one going from from high arousal, when there is an active response, to neutral arousal, when there is no active response. It seems unlikely that the arousal will drop into the negative range simply due to the absence of an active arousal. The BVP amplitude, then, can be seen as a crude measure of this part of the arousal dimension, where low values represents a spike in vasoconstriction, and the emotions associated with a sympathetic nervous system response to risk, and higher values indicate a lower vasoconstriction and a more calm, relaxed emotional state.

Determining valence, as an additional dimension to assess the emotional content with further accuracy, would be interesting, but is not practical to do from the BVP amplitude alone. However, due to the nature of the the "fight or flight" response and its area of utility, we can tentatively assume that the measured arousal has a certain negative valence.

Our implementation of emotions also has a componential quality, as the internal emotion can be said to be the result of an appraisal of the external environment, in this case represented by the game frames. The general strategy then is one combining certain aspects of appraisal methods, with what might be described as a hard-wired approach to emotion, represented by a strong, pre-determined connection between inputs and emotional states, as mediated by the trained neural networks.

In the *Visceral Machines* [46] project, which much of our project is modelled after, the researchers leveraged an assumed correlation between high arousal, as determined by the BVP amplitude, and dangerous situations for the self-driving agent, such as near collisions, in order to improve the performance of their DQN algorithm. The basic idea is that by receiving a negative reward when the agent is in a state that is likely to precede a collision, and a positive reward when the opposite is true, the agent is better able to learn how to avoid such situations.

In this thesis we theorise that a similar effect might be achieved in a gaming context. All though these environments are very different in many respects, there are also many key similarities between a gaming and a self-driving scenario. For instance, both require a large amount of focus on the task at hand, and thus benefit from high levels of concentration and engagement. Despite this, a driving scenario, with its higher associated risk, might ultimately be a more reliable elicitor of observable emotional reactions in humans, than a video game. There may however still be patterns of emotional reactions associated with risky situations in video games, that, all tough less extreme, can be beneficial to the development of an artificial agent. Furthermore, games, despite their lower stakes, are specifically designed to create engagement with their processes, thereby increasing the players investment in their performance and building an elevated sense of experienced risk or danger associated with bad results.

6.2 Integrating the BVP Amplitudes into the DQN Architecture

In order to integrate the BVP amplitudes into the DQN as an emotional signal, we created a custom reward function that calculates a new reward for each stored memory. The new reward function takes the BVP signal into consideration as an additional emotional reward. The emotional reward can be interpreted as an *intrinsic reward*, generated by the agents internal state, as opposed to the *extrinsic reward*, which is given by the external environment.

To estimate the emotional reactions we use the models we trained in chapter 4. For our experiments, we train several emotional DQN (EDQN) agents, each using a CNN that has been trained on a single participant. Therefore each agent represents the emotional reactions of a single person. This approach enables us to compare how the data of each participant affects the baseline algorithm, as well as create models that represents different emotional 'personalities'.

For each frame collected from the game environment the BVP model estimates the associated amplitude. The amplitude is given in a range from 0.1, representing the maximum recorded vasoconstriction and high arousal, to 1, indicating low vasoconstriction and arousal. The minimum value is 0.1 due to this being the minimum peak value used when calculating the BVP amplitudes to be used for training. For easier calculations, the predicted amplitude value is normalized to a range of 0 to 1, by use of the function expressed in Equation 6.1, where *amp* is the raw amplitude, *amp'* is the normalized amplitude, and *mpv* is the minimum peak value used in the preprocessing of the BVP amplitudes.

$$amp' = \frac{amp - mpv}{1 - mpv} \tag{6.1}$$

The normalized amplitude is further used for calculating an intrinsic reward that is given to the agent as an emotional signal. The intrinsic emotional reward per state, R_i, is given in Equation 6.2, where *amp'* is the normalized BVP amplitude.

$$R_i = (amp' - 0.5) \times 5 \tag{6.2}$$

This shapes the the intrinsic reward into the range of -5 to 5, making it comparable in size relative to the extrinsic reward already provided by the DQN algorithm. If the arousal is kept low, indicated by high amplitudes, this method gives a positive reward of up to a maximum 5, if the arousal levels rise above the relative middle mark, the reward turns to a negative one of up to -5. This will provide the agent with an incentive to keep the arousal low. If the correlation between high estimated arousal and risky situation in the environment holds, this could potentially help the agent to avoid more risky situations, in favour of safer options. This could further increase the training efficiency of the algorithm, especially at an early stage, where deep knowledge of the best policy is yet to be established.

Each memory stored by the DQN contains a set of four game frames that are stacked on top of each other, with the associated total reward for all frames. To obtain the associated emotional reward, we calculate the emotional reward for each game frame, adding them up for a total emotional reward for the set, before integrating it with the extrinsic reward.

A weighting variable was set between 0 and 1, to represent the degree to which the agent will favour the emotional reward over the extrinsic reward. This allows us to adjust the emotional emphasis of the agent. The total reward is calculated using the function in Equation 6.3, where R is the total, combined reward, R_e is the extrinsic reward, R_i is the total intrinsic emotional reward, and W is the weighting variable.

$$R = R_e \times (1 - W) + W \times R_i \tag{6.3}$$

This means that running the EDQN with W = 1 will be equivalent to training only on the emotional reward, while setting W = 0 is the same as running the *vanilla* DQN model. This scheme allows us to explore how different values of W affect the training process of the agent.

6.3 Experiments and Results

We ran several experiments to test the performance of different EDQN models in the SMB environment. In order to compare the models we used the same type of metrics that were applied to compare DQN models in chapter 5, namely the average scores achieved, where scores are measured as the total extrinsic reward achieved, and the finish rates, which gives the percentage of episodes where the agent was able to reach the end goal. Both the average scores and the finish rates are calculated over the last 1000 episodes. In this section, we go through the nature and results of these experiments, and briefly discuss the conclusions that can be drawn from these. It is important to note here that, due to the preliminary nature of these experiments, inaccuracies in measured performance may occur, as we only had time to run a single instance of each example model. Ideally, several instances of each model would be trained, and results would be averaged out, for a more accurate representation. Therefore, we should approach the data in an exploratory manner, looking for interesting observations, but being wary that more thorough investigations are in order to confirm any discovered patterns.

6.3.1 Training one EDQN per Participant

We ran experiments wich involved integrating the models of each participant in the dataset into a separate EDQN model, in order to determine how these perform compared to the *vanilla* DQN, as well as each other. For all these models the emotional weighting variable (W) was set to 0.5, meaning the extrinsic reward makes up half of the total reward, and the other half is given by the intrinsic signal. We ran the models for a total of 10 000 episodes each. Table 6.1 and 6.2 shows an initial comparison of these

Model	Episodes to reach average score of: 1000	1500	2000	2500	Max	Max score
Agent X: vanilla	1881	2955	4095	6027	8348	2850
Agent 0: p0	1566 (-315)	2225 (-730)	2840 (-1255)	4702 (-1325)	9912 (1564)	2801 (-49)
Agent 1: p1	1817 (-64)	2796 (-159)	3585 (-510)	4771 (-1256)	9952 (1604)	2780 (-70)
Agent 2: p2	1866 (-15)	2799 (-156)	3682 (-413)	5900 (-127)	9390 (1042)	2809 (-41)
Agent 3: p3	1899 (18)	2658 (-297)	3511 (-584)	5337 (-690)	9954 (1606)	2817 (-33)
Agent 4: p4	2275 (394)	3634 (679)	4333 (238)	5909 (-118)	7984 (-364)	2726 (-124)
Agent 5: p5	1778 (-103)	2671 (-284)	3589 (-506)	6217 (190)	7965 (-383)	2798 (-52)
Agent 6: p6	2029 (148)	2788 (-167)	3335 (-760)	4718 (-1309)	6566 (-1782)	2736 (-114)
Agent 7: p7	1945 (64)	2787 (-168)	3638 (-457)	4617 (-1410)	8872 (524)	2797 (-53)
Agent 8: p8	2362 (481)	3227 (272)	3903 (-192)	6395 (368)	9073 (725)	2719 (-131)
Agent 9: p9	1943 (62)	2605 (-350)	3260 (-835)	4786 (-1241)	9439 (1091)	2816 (-34)

Table 6.1: This table shows an overview of the amount of episodes needed by the emotional DQN models, for each participant and using a emotional weight of 0.5, to reach certain average scores, over total of 10 000 episodes. Averages are calculated over the last 1000 episodes.

Model	Episodes to reach finish rate of: 20%	40%	60%	80%	Max	Max finish rate
Agent X: vanilla	3670	4159	5405	6869	9091	0.85
Agent 0: p0	2834 (-836)	3226 (-933)	4238 (-1167)	8388 (1519)	9660 (569)	0.84 (-0.0)
Agent 1: p1	3404 (-266)	3905 (-254)	4520 (-885)	8288 (1419)	9989 (898)	0.83 (-0.02)
Agent 2: p2	3272 (-398)	3871 (-288)	5180 (-225)	7283 (414)	9638 (547)	0.84 (-0.01)
Agent 3: p3	2937 (-733)	3743 (-416)	4970 (-435)	9651 (2782)	9949 (858)	0.83 (-0.02)
Agent 4: p4	3920 (250)	4573 (414)	5638 (233)	NA	7975 (-1116)	0.8 (-0.05)
Agent 5: p5	3009 (-661)	3844 (-315)	5257 (-148)	7563 (694)	7961 (-1130)	0.83 (-0.02)
Agent 6: p6	3068 (-602)	3595 (-564)	4223 (-1182)	6562 (-307)	6633 (-2458)	0.8 (-0.04)
Agent 7: p7	3332 (-338)	3803 (-356)	4566 (-839)	8266 (1397)	8866 (-225)	0.83 (-0.01)
Agent 8: p8	3984 (314)	4467 (308)	5955 (550)	NA	9073 (-18)	0.78 (-0.06)
Agent 9: p9	2901 (-769)	3588 (-571)	4723 (-682)	7583 (714)	9529 (438)	0.85 (-0.0)

Table 6.2: This table shows an overview of the amount of episodes needed by the emotional DQN models, based on each each participant and using a emotional weight of 0.5, to reach certain finish rates, over total of 10 000 episodes. Finish rates are calculated over the last 1000 episodes.

models performance, in terms of the number of episodes needed to reach certain threshold values. The numeral ID's given to each model is the same as the one used for that participant in the dataset.

From the comparison of the average scores in Table 6.1 we see that, while certain models performed poorly compared to the *vanilla* DQN,

others were able to significantly outperform the standard version in certain respects.

Some particular observations of interest are that 9 out of 10 of EDQNs are able to reach an average score of 2000, and eight out of ten reach an average score of 2500, which is in the upper range of achievable scores, in fewer episodes than the standard DQN model. Some do this by significant margin, e.g, Agent 0, which achieves an average score of 2000 and 2500 using, respectively, 1255 and 1325 fewer episodes than the DQN. Others, e.g., Agents 4 and 5, offer less significant improvements. However, the fact that most of the EDQN models achieve comparable or better performance than the DQN indicates that the intrinsic signal, as it has been implemented, is a mostly positive influence on the early learning process of the model.

We also observe that the improvements associated with the intrinsic reward occur mostly in the beginning and middle of the learning process, while having a more negative effect towards the end. Many of the agents need more episodes to reach a max average score, and none of the agents reaches max average scores quite as high as the pure DQN agent. This effect is to be expected as it is a natural result of the emotional signal helping the agent to learn certain beneficial knowledge in early stages, while pure logical considerations become more important in the final stages of optimization. This is analogous to how such emotions may help in human learning, guiding early attempts and motivations while giving way to knowledge and experience over time.

The observations made from the average scores are further supported by the comparison of finish rates in Table 6.2. Here we see that eight of the ten EDQN agents where able to reach a finish rate of 60% in fewer episodes than the standard DQN, demonstrating the benefit of the intrinsic reward signal. We can here also more clearly see the reduced benefit of the intrinsic signal toward the later episodes, as only one of the EDQN agents are able to "beat" the standard DQN to an 80% finish rate, and none reach a higher maximum finish rate.

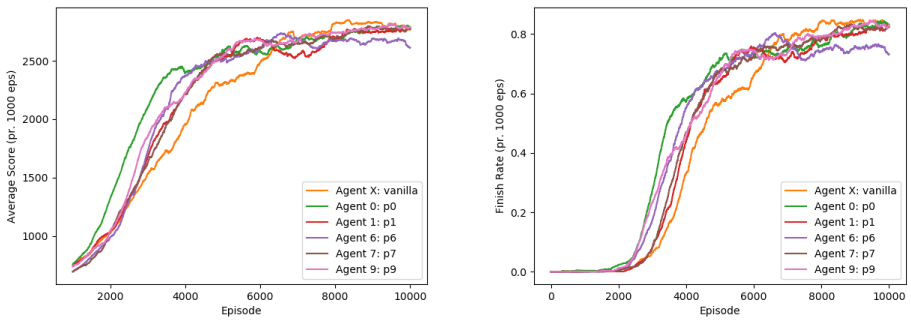

Figure 6.1: Comparing the average scores and finish rates of the top five EDQN agents over 10 000 episodes with an emotional weighting variable of 0.5.

For plot analysis, we separated the agents into two groups, containing the top and bottom five agents, based on their performances. Agents 0, 1, 6, 7, and 9 were deemed to be the top five agents, reaching an average score of 2500 and a 60% finish rate, in respective ranges of 1241 to 1420, and 682 to 1182, fewer episodes than the standard DQN. A plot comparing these agents can be seen in Figure 6.1. Conversely, the bottom five, namely agents 2, 3, 4, 5 and 8, reached average scores of 2500 and a finish rate of 60% in ranges of, respectively, 690 fewer to 368 more, and 435 fewer to 550 more, episodes than the standard DQN. A plot comparison these agents can be seen in Figure 6.2.

From the plot analysis we can more clearly see the observed effects, where most, or arguably all, of the models achieve better or comparable results to the standard DQN, up until the latter part of the training process. Standouts among the best performing agents are 0 and 6, both showing clear early improvements over the *vanilla* DQN agent. Among the bottom five models, only agent 4 and 8 show no early improvement over the standard DQN.

Another interesting observation is that the two agents that arguably did the best, namely Agent 0 and 6, were based on the participants with 'lots' of prior experience playing these types of games. This might be coincidental, but could also possibly indicate that models trained on players with more experience provide more useful contributions to the learning process. A possible reason for this could be that the prior knowledge of what constitutes risky game situations results in a more informative reaction pattern that is reflected in the data. This could be further investigated using an extended version of the current dataset.

In conclusion we show in these experiments that our emotional signal, recreated from the BVP amplitudes of certain participants, helps the DQN model to achieve increased learning performance per episode in the beginning and middle phases of learning to complete the level. These results are preliminary and more work remains in order to accurately assess how and why this improvement occurs, and to what degree this due to

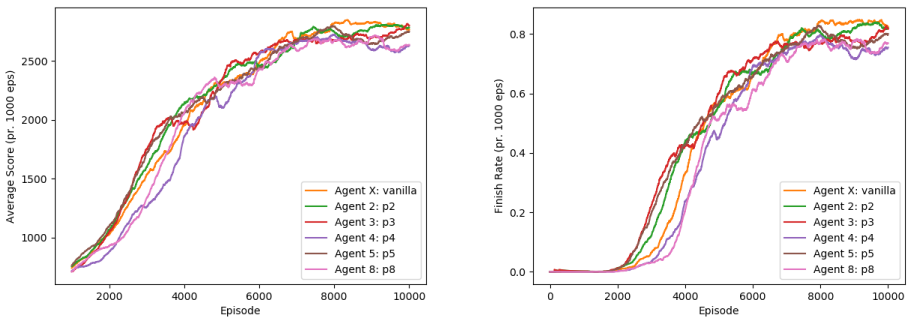

Figure 6.2: Comparing the average scores and finish rates of the bottom five EDQN agents over 10 000 episodes with an emotional weighting variable of 0.5.

the connection we have inferred between the sympathetic nervous system reactions and BVP, and risky situations in the game. However, the results are still a promising indication of the potential of our strategy to produce a useful emotion-based learning mechanism for reinforcement learning agents.

6.3.2 Testing different emotional weighting parameters

We tested some different values for the emotional weighting parameter W to see how this affected the performance of the EDQNs. We chose todo the test on the two best performing models, namely those of participant 0 and 6 from the previous section, to see if we could observed any specific changes in performance. We tested implementations of these models using 0.25 and 0.75, in addition to 0.5, as values for W. As explained in the previous section this is equivalent to using 25%, 50% and 75% emotional reward as part of the total reward associated with an experience. An overview of the performance of these implementations can be seen for participant 0 in Figure 6.3, and for participant 6 in Figure 6.4.

The most obvious observation from these experiments is that the increased W of 0.75 performs extraordinarily poorly for both sets of agents. At this rate the intrinsic reward seems to overwhelm the extrinsic reward, to a point where no real learning takes place. The implementations using a W of 0.25 was more successful in both cases; achieving slight improvements over the *vanilla* DQN for the participant 0 model, and outperforming the W=0.5 implementation for both participant in the late stages of training. The models using the previously explored W, equal to 0.5. do perform very well, and gives the best average results for both participant models.

From these observations we can conclude that the value of W certainly has an impact on the performance of the model, and that a too high value has a disastrous impact on the agents learning. Lower values seem to be more appropriate, and a W of 0.5 results in a decent amount of improvement in the early stages, while a lower W of 0.25 achieves

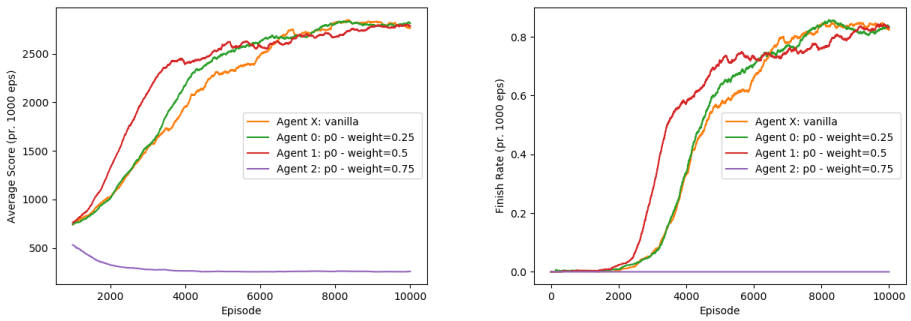

Figure 6.3: Comparing the average scores and finish rates of the EDQN model based on participant 0 over 10 000 episodes with emotional weighting variables of 0.25, 0.5 and 0.75.

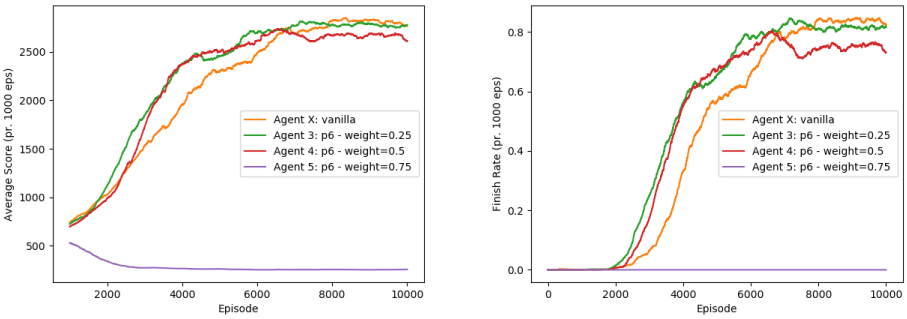

Figure 6.4: Comparing the average scores and finish rates of the EDQN model based on participant 6 over 10 000 episodes with emotional weighting variables of 0.25, 0.5 and 0.75.

better optimization towards the end. This is consistent with the idea that more focus on the analytical logic of the extrinsic reward is more beneficial towards the end of the current training process, where fine-tuned optimization is more important than broad and approximate reactions.

The observations documented here indicate that a more customized reward scheme is likely appropriate for our model, where the emotional signal becomes less influential as the agent acquires better knowledge. This may achieve the best of both worlds, retaining the early benefits of the intrinsic signal, while discounting it later for better optimization. In the next section make an initial attempt at implementing such a decaying intrinsic reward.

6.3.3 Testing a Decaying Intrinsic Reward

As we have discussed in this section, we believe that in the cases where a constant intrinsic signal seems to benefit learning, there is a struggle to optimize performance due to an increased need for pure analytical emphasis. If this is the case, a customized reward scheme that puts more weight on the intrinsic signal early, while putting more emphasis on the extrinsic reward in the later stages, could lead to an overall better learning performance. In this section we document an initial attempt at implementing such a customized reward scheme, using a decaying intrinsic reward.

We used the previously discussed EDQN model for participant 0, using a constant W=0.5, as a base and set the recorded reward to be calculated using a value for W that decays at a set rate with every episode. We tested two different rates of decay for W; 0.9996 and 0.9998. Table 6.3 shows the changes in W over 10 000 episodes.

The decay rate was implemented so that it is applied when the reward for a given memory is stored. A plot comparison of the EDQN models using a decaying W, with the one using a stable W of 0.5, as well as the *vanilla* DQN, can be seen in Figure 6.5. From this we can see that the currently implemented decay scheme does not seem to retain most of the

Values of W for different decay rates (DR)

Episode no:	DR: 0.9996:	DR: 0.9998:
0	0.5	0.5
1000	0.335	0.409
2000	0.225	0.335
3000	0.15	0.274
4000	0.101	0.224
5000	0.068	0.184
6000	0.045	0.150
7000	0.03	0.123
8000	0.02	0.101
9000	0.014	0.082
10000	0.009	0.067

Table 6.3: The percentages of the total reward that is given by the intrinsic emotion signal for various episode counts, using decay rates of 0.9996 and 0.9998.

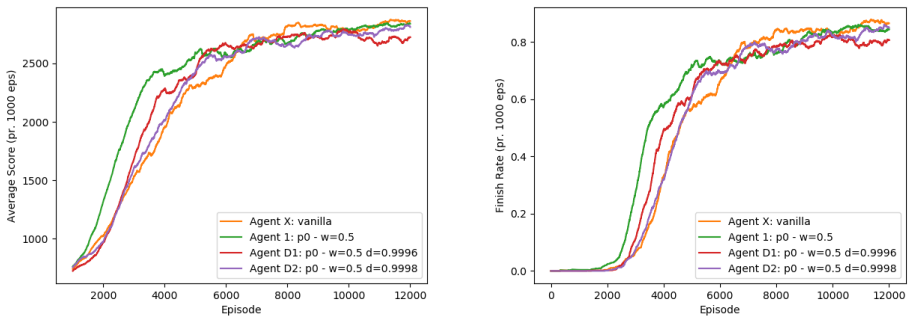

Figure 6.5: Comparing performance of EDQN models based on participant 0, using different decay rates for W, with one using a stable W of 0.5, as well as the "vanilla" DQN, over 12 000 episodes.

early learning benefits we associated with the intrinsic reward. Agent D1, using a decay rate of 0.9996 seems to do a bit better with this, while agent D2, using a decay of 0.9998, interestingly barely outperforms the *vanilla* DQN agent X in this respect. Neither agent D1 or D2 are able to achieve much improvement in terms of faster optimization either.

There are several possible reasons for why these experiments fail to show any improvement using a decaying W. Something to keep in mind is simply the aforementioned inaccuracies that must be an assumed possibility for the trained models. In this case a more thorough investigation, averaging out performances for several equivalent models, might reveal a alternative picture of the situation. Aside from this, one possibility is that the decay rate of W is to fast, or that it should start from a higher base value. This might help with retaining more of the early

improvements we see in the agent 1. This would however also be likely to further hinder later optimization.

An alternative scheme for the intrinsic reward could be to apply it at a higher value early while discounting it completely at some later point. It could also be that the models might perform better with a different kind decay and reward scheme than the one implemented here. An approach that might be more successful could be to store the intrinsic and extrinsic rewards as separate values in the replay buffer, calculating the functional reward before each training step. This would allow for greater control over W and make the reward for each memory independent of the decay rate at the time it was stored.

Many possibilities for more fine tuning of the W parameter and the reward scheme is certainly possible, in order to find the most ideal general, and model-particular, methods to apply in this case. Beyond the discussion here and in the final chapter, however, such investigations remain outside the scope of this particular thesis.

Chapter 7

Conclusion and Discussion

In this final chapter we will go through, evaluate and conclude the work produced in this thesis, as well as discuss possibilities for the future. We begin by summarizing the thesis, and the work produced therein. We then restate our problem statement and objectives, and discuss the main contributions made in this regard. We conclude with a general discussion on the significance of our contributions and findings, as well as possibilities for improvements of our methods and other ideas for future work.

7.1 Summary

For millennia, humans have sought the ability to create artificial entities that think, act and feel independently. In modern times, this goal is largely pursued in the form of AI, as well as advanced robotics. Even though much has been achieved in the field of AI, like the recent decades advances in ML and DL, many challenges must still be overcome in order to create a general artificial intelligence that can truly think and act independently. *Affective Computing* is a sub-field of AI that deals with the challenge of creating machines that can recognize, express and utilize emotions. There are good reasons to aim for this goal, including the observation, supported by findings in a range of related fields, that emotions are a crucial element of most of the important areas of human cognition, including many of those previously believed to be ruled by logic and rationality alone. This suggests that, if we are to create artificial minds that operate in a similar manner to humans, endowing them with certain affective abilities could be a necessary prerequisite. In a more immediate context, many current applications and artificial systems deal with emotions in various ways, such as social robots, chat-bots and virtual personal assistants, which will benefit from us improving our understanding of *Affective Computing*.

One of the more challenging goals of this field is to give machines the ability to have emotions of their own, in the sense that they regulate and utilize internal emotional states in its decision making processes. Many approaches have been used for finding ways to do this. In this thesis we mainly focus on the application of emotions in the field of RL. We are particularly inspired by a method used by McDuff and Kapoor [46],

where they collected data from real human sessions and used this to train a CNN to reproduce an emotional signal, which they then applied to a self-driving RL agent. In this thesis we further explored this general method by applying it to a new context, namely that of playing the classic video game Super Mario Bros.

In order to obtain the data needed for this, we developed a dataset which we named Toadstool. The dataset contains game, video and physiological data from ten participants playing the game. We also created the software tools and protocols necessary to collect, process and extend the dataset. The dataset is openly available for research purposes and the associated paper was published at the 11th ACM Multimedia Systems Conference [70]. The paper has also been attached to the appendix of the thesis.

We then processed the BVP data by normalization and detection of peak values which were spread across the sample collection. The resulting data was used to train a series CNN model to predict the BVP amplitude based on frames from the game. The constriction of BVP amplitude is associated with an immediate "fight or flight" response from the sympathetic nervous system. The idea is that this emotional signal might assist an artificial agent to recognize certain situations, like those that represent an immediate risk in the game. The data was used train the model to estimate such an emotional response. The trained CNN models achieve a certain accuracy, but still leaves much to be desired in terms of reproducing an informative signal. Still, the prediction accuracy achieved might suffice to create a directed influence on a RL agent.

We built a *vanilla* DQN to serve as a baseline for introducing the intrinsic emotion, and tested different parameters for the DQN to find a setup and performance level suitable for our purposes. The CNN models predicting the BVP amplitudes for each participant, were then introduced into the DQN through a custom reward scheme, creating a set of EDQNs. We ran experiments where we compared the performance of the EDQN models versus the *vanilla* DQN model. From these we observe that while some of the models perform poorly, and all expectantly struggle with the final optimization, some show a significant early improvement over the standard version, indicating certain benefits of using the intrinsic signal for these instances. We also tested some different prioritisation schemes for the best performing models, placing varying weight on the intrinsic reward in relation to the extrinsic reward. These experiments demonstrate that putting too much focus on the intrinsic signal is disastrous for the learning process, at least given our current setup. The results of our experiments also indicate that there might be a benefit to applying our emotional signal early in the learning process, while the emotion may hinder optimization in later stages. Therefore, we attempted to implement a custom reward scheme using a decaying emotional weighting variable, in order to achieve an overall better performing model, however, these models failed to show any significant improvement. All though there could be many reasons for this, there is a high likelihood that the lack of improvement is due to

our specific implementation, and that a more appropriate scheme might achieve better results.

In summary, over the course of this thesis we have produced and applied the necessary prerequisites to collect relevant data and perform experiments on emotional reinforcement learning agents playing SMB. We have created and demonstrated a pipeline for introducing emotion into a standard DQN model playing a video game, and methods for performing experiments with the resulting EDQNs. From our experiments we can tentatively conclude that, despite certain limitations in our performed process, the intrinsic emotion seem to grant certain benefits to the performance of the system. More investigation is in order of determine the exact nature and origin of this improvement, and whether it is actually the result of a connection between the collected emotional data and specific game situations, like has been theorized in this work.

7.2 Main Contributions

In this thesis we set out to provide an answer to the following question:

> *Can human emotions be used to improve the performance, or otherwise change the behaviour, of an autonomous agent playing video games?*

In chapter 1 we set three objectives which, when completed would bring us closer to an answer to this question. The main contributions of this thesis are contained within the work done to fulfill these three objectives. In this section we reiterate the objectives and show how they are supported by our work.

Objective 1 Develop a dataset for the purposes of conducting research involving the physical state, signals and behaviour, related to emotional states, of humans playing a video game. The dataset should contain; game data from human play sessions, correlated physiological and/or other observational data, and other information relevant to analysing the data.

This objective is supported by the development of the Toadstool dataset, which contains game, video and physiological data from humans playing SMB.

Objective 2 Train a deep neural network to predict an emotionally correlated human signal based on the collected dataset.

This objective is supported by the training of CNN models to predict BVP amplitudes, using the BVP data from the Toadstool dataset.

Objective 3 Build a model of a reinforcement learning agent that plays the same video game and perform experiments to show how its learning and behaviour is affected by introducing the emotional signal as an intrinsic reward mechanism.

This objective is supported by the development of a baseline DQN model, and the following integration of the BVP based CNNs to create EDQN models, as well as the set of experiments performed with these models, and the observations made from the results.Based on this work, the main contributions of this thesis, to the field and towards an answer to the question posed above, can be summarized in the two following points.

Firstly, we have presented the open dataset Toadstool [70], as well as tools for its collection, synchronization and extension. This enables the rest of our research, and also provides a resource to the community for further investigation and experimentation into the field of *Affective Computing*. Secondly, we have presented and explored a pipeline for using this dataset to perform research on a specific context of emotion in RL. We have performed experiments that show some promising results, and indicate further possibilities for developing and fine-tuning this pipeline.

Even though these contributions do not conclusively answer our initial research question, they do provide some interesting and relevant observations, as well as tools and paths for further work, that may assist in the search for such an answer.

7.3 Discussion: Evaluation, Improvements and Ideas for Future Work

Due to the exploratory and experimental nature of much of this thesis, many lessons and possibilities for improving the process were discovered along the way or revealed by the clarity of hindsight. In this final section we will further discuss and evaluate our project and its limitations, as well as some possible improvements, alternative strategies and possibilities for future work.

7.3.1 Data collection and processing

One area that has potential for improvement is the data collection and the processing of the BVP data into amplitudes. The BVP measurements is somewhat sensitive to distortion from movement. In retrospect it might have been a good idea to inform participants to keep as still as possible when playing, all though this might affect other data as well, for instance posture data obtained through video analysis. As the data from the E4 Accelerometer is included in the dataset, it is possible to correct for such distortions, which could end up improving the quality of the training data.

In processing the BVP data from the dataset we also used a standard peak detection algorithm that is not ideal for the purposes of determining the most informative BVP amplitude samples. One problem that we mentioned in chapter 4, is that the presence of extreme outliers in the the peak values might skew the real range and proportionality of the training data. This, as well as the general accuracy of the training data, could be improved by applying a custom peak detection algorithm.

7.3.2 Training and evaluating CNN models

As the CNN models we trained in chapter 4 only achieves a certain level of predictive power, there remains an opportunity to train these using other variations and techniques, that may lead to more reliable predictions and better emotional signals. Possibilities here include; adjusting the learning rate and other parameters, making changes to the layers of the CNN model, like the number and parameters of the convolutional layers, splitting or combing data in alternate ways, or applying other advanced deep learning techniques. As we a mentioned above, there is also the possibility of improving the quality of the training data, which could lead to improved performance in the CNN.

Another method that could be applied to possibly produce models with more predictive power than the current CNNs, is a multimodal approach, like using the BVP data together with additional data sources from the dataset. EDA and/or video data would be good candidates for this. These sources could be combined to produce greater predictive power, or used in conjunction to create a combinatory emotional signal, or even multiple ones.

A more thorough analysis of the trained models might also yield positive results. In chapter 4 we show a somewhat cursory analysis of the CNN prediction of some test images, however, there remains many interesting avenues to explore here. This could be done, for instance, by collecting more specific classes of test images and comparing how the models classify them, in order to assess what game situations, like jumps, enemies, levels etc., convert to low or high arousal values, and also how the various models align on these decisions. Many possibilities exist here, and gaining a better understanding of how the models actually function would make many things easier, like building customized reward schemes around the behaviour of specific models.

7.3.3 Implementing and evaluating the EDQN models

The culmination of our project is the integration of the trained CNNs into a DQN model to create EDQNs. Obviously, many of the previously discussed possible improvements to the previous processes, should ultimately improve the quality of the EDQN models. Beyond this, there are several aspects of the process of developing our EDQNs that could likely benefit from a more thorough approach, which we failed to apply, either because of lacking the necessary foresight and experience, or as a result of the need to limit the scope of the thesis due to time constraints. Additionally, there are many alternative approaches to various facets of our implementation, that could potentially yield better, or otherwise interesting, results.

First off, we chose to evaluate the models based solely on the metrics of average extrinsic reward achieved per episode (score), and finish rate. Other metrics, or more specific behavioral requirements, could be applied for a more detailed evaluation, depending on one's approach. We also only had the opportunity to run a single instance of each model, while a more

ideal experiment would have included running several instances of each implementation and averaging out results.

Furthermore, there are a plethora of different ways to integrate the predicted BVP amplitudes into the reward function of the DQN. In addition to alternate schemes, for instance using different conversions from prediction to intrinsic reward, the emotional weighting variable (W) can be manipulated in various ways to guide the learning process. Some examples of this are implemented in this thesis, however, many more possibilities exist here, for instance for a more effective decaying W.

Additionally, there are still open questions regarding the degree to which the effect of the eventual intrinsic reward is analogous to the one proposed by our theoretical framework, and more investigations is needed to draw any definitive conclusions about this, as well as the general efficacy of applying emotion in our given context. For instance, considering the actual prediction values produced by the CNN, as shown in Table 4.3 on page 61, it is a possibility that some part of the effect is due to the signal taking the form of a relatively stable negative reward for some agents, which could provide an incentive for some early beneficial behaviour.

Finally it seems prudent to question to what degree we have actually achieved an emotional intelligent machine with the EDQN. Certainly, if deep neural network based agents can be said to possess a degree of logical intelligence, then this is also present in our models, but can they be said to possess emotional intelligence? As stated above, more investigations is needed to determine how correlated the reproduced emotion is to actual human emotion, however, at the very least, the intrinsic reward can be viewed as a non-random, non-logic based, artificial affect, that introduces an element of intuition and personality to an otherwise purely logical machine. If our theoretical framework is in fact valid, and the theorized correlations are maintained in our process, some of our EDQNs have been shown to utilize this emotional quality, in conjunction with their logical abilities, to produce an increased ability for mastering their environment in a certain context. This, in our opinion, is sufficient to claim that, however rudimentary, these models do in fact exhibit a certain degree of emotional intelligence.

It is our hope that this thesis, and the work provided herein, may prove a contribution to the overall quest of creating more advanced artificial emotional intelligence. We also hope that this quest shall ultimately prove to be a worthy one for humanity, and not a path that will culminate in the development of technology to be used for manipulation and exploitation, or worse yet, some dark science fiction-like scenario, where humans suffer or perish at the mechanical hands of super-intelligent, emotional machines.

Bibliography

[1] Darren Abramson. "Descartes' influence on Turing." In: *Studies in History and Philosophy of Science* 42 (Dec. 2011), pp. 544–551. DOI: 10.1016/j.shpsa.2011.09.004.

[2] Ralph Adolphs, Simon Baron-Cohen, and Daniel Tranel. "Impaired Recognition of Social Emotions following Amygdala Damage." In: *Journal of Cognitive Neuroscience* 14.8 (2002), pp. 1264–1274. DOI: 10.1162/089892902760807258.

[3] *alexa*. https://developer.amazon.com/alexa/. Accessed: 2020-04-14.

[4] *Artbreeder*. https://www.artbreeder.com/. Accessed: 2020-04-14.

[5] J. Barnes. *The Complete Works of Aristotle*. Bollingen series v.2 *Politics, Book 1, Part 4*. Princeton University Press, 1995, p. 1989. ISBN: 9780691099507.

[6] G. Bradski. "The OpenCV Library." In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] Cynthia Breazeal. "Emotion and Sociable Humanoid Robots." In: *International journal of human-computer studies* 59.1–2 (July 2003), pp. 119–155. ISSN: 1071-5819. URL: https://doi.org/10.1016/S1071-5819(03)00018-1.

[8] Greg Brockman et al. *OpenAI Gym*. 2016. eprint: arXiv:1606.01540.

[9] John-John Cabibihan et al. "Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism." In: *International journal of social robotics* 5.4 (2013), pp. 593–618. DOI: 10.1007/s12369-013-0202-2.

[10] Garry Choy et al. "Current Applications and Future Impact of Machine Learning in Radiology." In: *Radiology* 288 (June 2018), p. 171820. DOI: 10.1148/radiol.2018171820.

[11] G. Cybenko. "Approximation by superpositions of a sigmoidal function." In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274.

[12] Shaundra B. Daily et al. "Chapter 9 - Affective Computing: Historical Foundations, Current Applications, and Future Trends." In: *Emotions and Affect in Human Factors and Human-Computer Interaction*. Ed. by Myounghoon Jeon. San Diego: Academic Press, 2017, pp. 213–231. DOI: 10.1016/B978-0-12-801851-4.00009-4.

[13] Antonio R. Damasio. *Descartes error : emotion, reason, and the human brain*. New York : G.P. Putnam, 1994. URL: https://search.library.wisc.edu/catalog/999764511802121.

[14] H. Damasio et al. "The return of Phineas Gage: clues about the brain from the skull of a famous patient." In: *Science* 264 (1994), pp. 1102–1105. DOI: 10.1126/science.8178168.

[15] Michael E. Dawson, Anne M. Schell, and Diane L. Filion. "The Electrodermal System." In: *Handbook of Psychophysiology*. Ed. by John T. Cacioppo, Louis G. Tassinary, and Gary G.Editors Berntson. 4th ed. Cambridge Handbooks in Psychology. Cambridge University Press, 2016, pp. 217–243. DOI: 10.1017/9781107415782.010.

[16] Benedetto De Martino et al. "Frames, biases, and rational decision-making in the human brain." In: *Science* 313.5787 (2006), pp. 684–687.

[17] Peter Denning et al. "Computing as a discipline." In: *Computer* 22 (Mar. 1989), pp. 63–70. DOI: 10.1109/2.19833.

[18] P. Ekman et al. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. Series in affective science. Oxford University Press, 1997. ISBN: 9780195104462.

[19] Paul Ekman et al. "Universals and cultural differences in the judgments of facial expressions of emotion." In: *Journal of personality and social psychology* 53.4 (1987), p. 712.

[20] Ch Fere. "Note sur des modifications de la resistance electrique sous l'influence des excitations sensorielles et des motions." In: *Compt Rend Soc de Biol* 8 (1888), pp. 217–219.

[21] Nico H Frijda, Peter Kuipers, and Elisabeth Ter Schure. "Relations among emotion, appraisal, and emotional action readiness." In: *Journal of personality and social psychology* 57.2 (1989), p. 212.

[22] Juley Anna Fulcher. "Vocal affect expression as an indicator of affective response." In: *Behavior Research Methods, Instruments, & Computers* 23 (1991), pp. 306–313. DOI: 10.3758/BF03203384.

[23] Sandra Clara Gadanho and John Hallam. "Robot Learning Driven by Emotions." In: *Adaptive Behavior* 9.1 (2001), pp. 42–64. DOI: 10.1177/105971230200900102.

[24] M. Garbarino et al. "Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition." In: *ICWMCHM 2014*. 2014, pp. 39–42.

[25] Ian Goodfellow et al. "Generative adversarial nets." In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[26] Joshua D. Greene et al. "An fMRI Investigation of Emotional Engagement in Moral Judgment." In: *Science* 293.5537 (2001), pp. 2105–2108. DOI: 10.1126/science.1062872.

[27] Joshua Greene and Jonathan Haidt. "How (and where) does moral judgment work?" In: *Trends in Cognitive Sciences* 6.12 (2002), pp. 517–523. DOI: 10.1016/S1364-6613(02)02011-9.

[28] Jiuxiang Gu et al. "Recent advances in convolutional neural networks." In: *Pattern Recognition* 77 (2018), pp. 354–377. DOI: 10.1016/j.patcog.2017.10.013.

[29] Hado V Hasselt. "Double Q-learning." In: *Advances in neural information processing systems*. 2010, pp. 2613–2621.

[30] Gary Hatfield. "René Descartes." In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford University, 2018.

[31] Michael Hechter and Satoshi Kanazawa. "Sociological rational choice theory." In: *Annual review of sociology* 23.1 (1997), pp. 191–214.

[32] Seyedmehdi Hosseinimotlagh and Evangelos E Papalexakis. "Unsupervised content-based identification of fake news articles with tensor decomposition ensembles." In: *Proceedings of the Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*. 2018.

[33] Clark L Hull. *Principles of behavior*. Vol. 422. Appleton-century-crofts New York, 1943.

[34] M. Idel and State University of New York. *Golem: Jewish Magical and Mystical Traditions on the Artificial Anthropoid*. SUNY Series in Judaica: Hermeneutics, Mysticism, and Religion Series. State University of New York Press, 1990. ISBN: 9780791401606.

[35] Younbo Jung and Kwan Min Lee. "Effects of physical embodiment on social presence of social robots." In: *Proceedings of PRESENCE* (2004), pp. 80–87.

[36] Krisztian Kasos et al. "Does the electrodermal system "take sides" when it comes to emotions?" In: *APBF Journal* 43.3 (2018), pp. 203–210.

[37] Christian Kauten. *Super Mario Bros for OpenAI Gym*. https://github.com/Kautenja/gym-super-mario-bros. 2018.

[38] Hyoung-Rock Kim and Dong-Soo Kwon. "Computational Model of Emotion Generation for Human–Robot Interaction Based on the Cognitive Appraisal Theory." In: *Journal of Intelligent & Robotic Systems* 60.2 (Nov. 2010), pp. 263–283. DOI: 10.1007/s10846-010-9418-7.

[39] Derek A Laffan et al. "The relationships between the structural video game characteristics, video game engagement and happiness among individuals who play video games." In: *Computers in Human Behavior* 65 (2016), pp. 544–549.

[40] Richard S Lazarus. "Cognition and motivation in emotion." In: *American psychologist* 46.4 (1991), p. 352.

[41] Yibo Li, Liangren Zhang, and Zhenming Liu. "Multi-objective de novo drug design with conditional graph generative model." In: *Journal of cheminformatics* 10.1 (2018), p. 33.

[42] R. Liu and J. Zou. "The Effects of Memory Replay in Reinforcement Learning." In: *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2018, pp. 478–485.

[43] G. F. Loewenstein et al. "Risk as feelings." In: *Psychological Bulletin* 127.2 (2001), pp. 267–286. DOI: 10.1037/0033-2909.127.2.267.

[44] Atsushi Matsuda, Hideaki Misawa, and Keiichi Horio. "Decision making based on reinforcement learning and emotion learning for social behavior." In: *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE. 2011, pp. 2714–2719.

[45] Laurie Kelly McCorry. "Physiology of the autonomic nervous system." In: *American journal of pharmaceutical education* 71.4 (Aug. 2007), pp. 78–78. DOI: 10.5688/aj710478.

[46] Daniel McDuff and Ashish Kapoor. "Visceral Machines: Risk-Aversion in Reinforcement Learning with Intrinsic Physiological Rewards." In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. Apr. 2019.

[47] Tom M Mitchell et al. *Machine learning*. 1997.

[48] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. "Emotion in reinforcement learning agents and robots: a survey." In: *Machine Learning* 107.2 (2018), pp. 443–480.

[49] Ziad Obermeyer and Ezekiel J Emanuel. "Predicting the future—big data, machine learning, and clinical medicine." In: *The New England journal of medicine* 375.13 (2016), p. 1216.

[50] Hussein Al Osman and Tiago H. Falk. "Multimodal Affect Recognition: Current Approaches and Challenges." In: *Emotion and Attention Recognition Based on Biological Signals and Images*. Ed. by Seyyed Abed Hosseini. Rijeka: IntechOpen, 2017. Chap. 5. DOI: 10.5772/65683.

[51] Manh Cuong Pham et al. "A clustering approach for collaborative filtering recommendation using social network analysis." In: *J. UCS* 17.4 (2011), pp. 583–604.

[52] Rosalind W. Picard. *Affective Computing*. Cambridge, MA, USA: MIT Press, 1997. DOI: 10.7551/mitpress/1140.001.0001.

[53] Rosalind W. Picard, Szymon Fedor, and Yadid Ayzenberg. "Multiple Arousal Theory and Daily-Life Electrodermal Activity Asymmetry." In: *Emotion Review* 8.1 (2016), pp. 62–75.

[54] A.R. Maxwell-Hyslop Pierre Grimal Stephen Kershaw. "Cadmus." In: *The Penguin Dictionary of Classical Mythology*. Penguin, 1992. ISBN: 9780140512359.

[55] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65 (1958), pp. 386–408. DOI: 10.1037/h0042519.

[56] Philipp V Rouast et al. "Remote heart rate measurement using low-cost RGB face video: a technical literature review." In: *Frontiers of Computer Science* 12.5 (2018), pp. 858–872.

[57] Sebastian Ruder. "An overview of gradient descent optimization algorithms." In: *ArXiv* abs/1609.04747 (2016).

[58] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors." In: *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988, pp. 696–699. ISBN: 0262010976.

[59] James A Russell. "A circumplex model of affect." In: *Journal of personality and social psychology* 39.6 (1980), p. 1161.

[60] James A Russell. "Evidence of convergent validity on the dimensions of affect." In: *Journal of personality and social psychology* 36.10 (1978), p. 1152.

[61] Jyotirmay Sanghvi et al. "Automatic Analysis of Affective Postures and Body MotionPhoebe to Detect Engagement with a Game Companion." In: *Proceedings of the 6th International Conference on Human-Robot Interaction*. HRI '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 305–312. DOI: 10.1145/1957656.1957781.

[62] J. Scheirer et al. "Frustrating the user on purpose: a step toward building an affective computer." In: *Interacting with Computers* 14.2 (2002), pp. 93–118.

[63] John Scott. "Rational choice theory." In: *Understanding contemporary society: Theories of the present* 129 (2000), pp. 671–85.

[64] Heung-yeung Shum, Xiao-dong He, and Di Li. "From Eliza to XiaoIce: challenges and opportunities with social chatbots." In: *Frontiers of Information Technology & Electronic Engineering* 19.1 (2018), pp. 10–26. DOI: 10.1631/FITEE.1700826.

[65] David Silver et al. "Mastering the game of Go with deep neural networks and tree search." In: *nature* 529.7587 (2016), p. 484.

[66] David Silver et al. "Mastering the game of go without human knowledge." In: *Nature* 550.7676 (2017), pp. 354–359.

[67] S. Singh et al. "Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective." In: *IEEE Transactions on Autonomous Mental Development* 2.2 (2010), pp. 70–82.

[68] *Siri*. https://www.apple.com/ios/siri/. Accessed: 2020-04-14.

[69] Craig A Smith and Phoebe C Ellsworth. "Patterns of cognitive appraisal in emotion." In: *Journal of personality and social psychology* 48.4 (1985), p. 813.

[70] Henrik Svoren et al. "Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros." In: *Proceedings of the 11th ACM Multimedia Systems Conference*. MMSys '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 309–314. ISBN: 9781450368452. DOI: 10.1145/3339825.3394939.

[71] Penelope Sweetser and Peta Wyeth. "GameFlow: a model for evaluating player enjoyment in games." In: *Computers in Entertainment (CIE)* 3.3 (2005), pp. 3–3.

[72] Kevin K. Tremper. "Pulse Oximetry." In: *CHEST* 95.4 (Apr. 1989), pp. 713–715. DOI: 10.1378/chest.95.4.713.

[73] Hado Van Hasselt, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." In: *Thirtieth AAAI conference on artificial intelligence*. 2016.

[74] Juan Velásquez. "Modeling emotion-based decision-making." In: *Emotional and intelligent: The tangled knot of cognition* (1998), pp. 164–169.

[75] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[76] Thurid Vogt, Elisabeth André, and Johannes Wagner. "Automatic Recognition of Emotions from Speech: A Review of the Literature and Recommendations for Practical Realisation." In: *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*. Ed. by Christian Peter and Russell Beale. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 75–91. DOI: 10.1007/978-3-540-85099-1_7.

[77] Kazuyoshi Wada and Takanori Shibata. "Living with seal robots—its sociopsychological and physiological influences on the elderly at a care house." In: *IEEE transactions on robotics* 23.5 (2007), pp. 972–980.

[78] Kazuyoshi Wada and Takanori Shibata. "Robot therapy in a care house-its sociopsychological and physiological effects on the residents." In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE. 2006, pp. 3966–3971.

[79] Kazuyoshi Wada and Takanori Shibata. "Robot therapy in a care house-results of case studies." In: *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE. 2006, pp. 581–586.

[80] Xintao Wang et al. "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks." In: *CoRR* abs/1809.00219 (2018). arXiv: 1809.00219. URL: http://arxiv.org/abs/1809.00219.

[81] Christopher JCH Watkins and Peter Dayan. "Q-learning." In: *Machine learning* 8.3-4 (1992), pp. 279–292.

[82] Gillian M. Wilson and M. Angela Sasse. "Do Users Always Know What's Good For Them? Utilising Physiological Responses to Assess Media Quality." In: *People and Computers XIV — Usability or Else!* Ed. by Sharon McDonald, Yvonne Waern, and Gilbert Cockton. London: Springer London, 2000, pp. 327–339. ISBN: 978-1-4471-0515-2.

[83] Wilhelm Wundt. "Outlines of psychology (1897)." In: *Toronto: York University* (2004).

[84] C. Yu et al. “Emotional Multiagent Reinforcement Learning in Spatial Social Dilemmas.” In: *IEEE Transactions on Neural Networks and Learning Systems* 26.12 (2015), pp. 3083–3096.

[85] Li Zhou et al. “The design and implementation of xiaoice, an empathetic social chatbot.” In: *Computational Linguistics* 46.1 (2020), pp. 53–93.

Appendices

Paper I — Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros

Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros

Henrik Svoren*
henrik.svo@gmail.com
University of Oslo
Oslo, Norway

Vajira Thambawita‡
vajira@simula.no
SimulaMet
Oslo, Norway

Pål Halvorsen‡
paalh@simula.no
SimulaMet
Oslo, Norway

Petter Jakobsen†
peja@helse-bergen.no
NORMENT, Haukeland University Hospital
Bergen, Norway

Enrique Garcia-Ceja
enrique.garcia-ceja@sintef.no
SINTEF Digital
Oslo, Norway

Farzan Majeed Noori
farzanmn@ifi.uio.no
University of Oslo
Oslo, Norway

Hugo L. Hammer*
hugoh@oslomet.no
OsloMet
Oslo, Norway

Mathias Lux
mlux@itec.aau.at
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria

Michael Alexander Riegler§
michael@simula.no
SimulaMet
Oslo, Norway

Steven Alexander Hicks‡§
steven@simula.no
SimulaMet
Oslo, Norway

ABSTRACT

Games are often defined as engines of experience, and they are heavily relying on emotions, they arouse in players. In this paper, we present a dataset called *Toadstool* as well as a reproducible methodology to extend on the dataset. The dataset consists of video, sensor, and demographic data collected from ten participants playing *Super Mario Bros*, an iconic and famous video game. The sensor data is collected through an Empatica E4 wristband, which provides high-quality measurements and is graded as a medical device. In addition to the dataset and the methodology for data collection, we present a set of baseline experiments which show that we can use video game frames together with the facial expressions to predict the blood volume pulse of the person playing Super Mario Bros. With the dataset and the collection methodology we aim to contribute to research on emotionally aware machine learning algorithms, focusing on reinforcement learning and multimodal data fusion. We believe that the presented dataset can be interesting for a manifold of researchers to explore exciting new interdisciplinary questions.

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Computing methodologies** → *Visual inspection*; *Neural networks*; *Classification and regression trees.*

KEYWORDS

Multimedia Datasets, Neural Networks, Emotional Machines, Machine Learning

ACM Reference Format:
Henrik Svoren, Vajira Thambawita, Pål Halvorsen, Petter Jakobsen, Enrique Garcia-Ceja, Farzan Majeed Noori, Hugo L. Hammer, Mathias Lux, Michael Alexander Riegler, and Steven Alexander Hicks. 2020. Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros. In *11th ACM Multimedia Systems Conference (MMSys'20), June 8–11, 2020, Istanbul, Turkey.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3339825.3394939

1 INTRODUCTION

"Stop Dave. Stop Dave. I am afraid. I am afraid Dave." This iconic quote from Stanley Kubrick's *2001: A Space Odyssey* is taken from a scene where the sentient computer system HAL 9000 is pleading for life as the human operator is about to shut it down. The movie was released in 1969, and looking at the state of artificial intelligence

*Also affiliated with SimulaMet, Norway
†Also affiliated with University of Bergen, Norway
‡Also affiliated with Oslo Metropolitan University, Norway
§These authors share the senior position of the article.

MMSys'20, June 8–11, 2020, Istanbul, Turkey

ACM ISBN 978-1-4503-6845-2/20/06…$15.00
https://doi.org/10.1145/3339825.3394939

Figure 1: Frames are taken from each of the 32 levels contained within *Super Mario Bros.* Note that each image is taken from the very first frame of each level. Levels in *Super Mario Bros.* are organized in groups of four and called worlds, so the first level is world 1-1, the second level is world 1-2, the fifth level is world 2-1, etc.

(AI) today, we can make two observations. First, people in the 60s and 70s were very optimistic about the future capabilities of AI. Second, we are far away from anything near the emotional intelligence that HAL 9000 expresses throughout the movie. For the most part, current AI systems are focused on performing well on specific tasks like classification, object detection or regression, while a machine that can express general intelligence is still far off. This is not negative in and of itself [7], but it is quite different from what people in the past imagined AI would be in the future, and what we might imagine today.

Using machine learning to interpret or detect human emotions is a growing field of research. This is commonly done using different types of media, such as images [2], sensor data [11], text [23], or some combination of the three [4, 14]. Recent works in this field have also moved to look at how human emotion data may affect the training and performance of deep learning algorithms. McDuff et al. [17] explore how human emotional response may affect the performance of a self-driving agent trained in a simulated environment. They showed that adding human-like signals, such as the blood volume pulse (BVP), helped improve the driving performance of the algorithm. The idea of supplementing today's machines with emotional or physiological signals is supported by the large amount of literature that shows that pure rational decision making is often not optimal in humans [3, 9, 18, 22]. Prior research shows that emotional content can help guide the decision-making process as well as make it more efficient [16]. Some early work also tried to use this for artificial agents [10]. Such findings suggest the possibility that similar benefits might be had by artificial agents, especially when engaged in human-like tasks or behavior.

Inspired by the work done by McDuff et al. [17], we look at other areas where the same principles may be applied, which in this case, is playing the well-known classical video game *Super Mario Bros.* While this game is not representative for all video games that are available right now, it is commonly accepted as a well-known, good example for a video game and can be considered representative for the jump and run and the arcade game genres. To perform experiments in that direction, we first need a dataset that contains both the frames from *Super Mario Bros.* and the sensory output of the player. As no such dataset exists, we collected gameplay data, sensor data, and facial expression data from ten different participants. Furthermore, we also made the dataset and all sources to re-produce the games played publicly available. We think this dataset is of great interest to many research communities as it consists of multiple modalities and is applied to a unique use case. The contributions of this paper are three-fold:

(1) We present a publicly available, multimodal dataset which focuses on the human component of intelligent machines along with a reproducible methodology to extend the dataset with additional data collection.
(2) We present a set of baseline experiments that aim to show how the dataset can be used to predict specific sensor values using a combination of data from the video game and facial expressions.
(3) We outline future applications and interesting research questions using the dataset.

To the best of our knowledge, this is the first open dataset that provides the (i) video frames of a person's facial expressions, (ii) the sensory output of the person playing a game, and (iii) data from the video game synchronized with the facial expressions and sensor data. The dataset opens up for a wide range of new and interesting analyses, and a proper and fair comparison between different methods, both from a psychological and a multimedia perspective. In the following, the process of collecting the data, as well as the resulting data, are described. Moreover, a baseline evaluation is presented, including suggestions for future research directions using the dataset.

ID	Age	Sex	Dominant hand	Hours per week	Years active	Prior experience	Game score
0	26	Male	Right	4-8	22	Lots	17, 100
1	48	Male	Left	0-1	1	Little	3, 000
2	28	Male	Right	0-1	0	None	300
3	32	Male	Right	4-8	4	Some	13, 300
4	32	Female	Right	0-1	5	Some	6, 400
5	30	Female	Right	0-1	5	Little	2, 700
6	35	Male	Left	1-4	30	Lots	14, 300
7	34	Female	Right	1-4	14	Some	3, 800
8	31	Female	Right	0-1	2	Little	200
9	27	Female	Right	0-1	5	Little	10, 600

Table 1: This table shows an overview of all participants included in the dataset.

2 DATA COLLECTION

The dataset was collected at our research laboratory located in Oslo, Norway. Participants were selected based on a set of criteria, mostly focused on their prior gaming experience. We wanted to collect data from people with a wide range of different game experience backgrounds. This includes those who have barely touched a video game to those who have been playing since childhood. Furthermore, we aimed to collect data from a balanced set of genders, meaning an even split of male and female participants. Each participant was asked to fill out a short questionnaire about their previous video game experience in addition to some information about themselves. An overview of the answers can be seen in Table 1. In total, ten participants were selected for the study, where each participant provided a written form of consent, allowing for their video, gameplay data, and sensor data to be shared openly for research and teaching purposes under the license Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)[1]. The dataset can be accessed via (https://datasets.simula.no/toadstool) or (https://osf.io/qrkcf/).

As for collecting the gameplay data, we developed a protocol that describes what data should be collected and how. This protocol went through multiple iterations as we performed a preliminary test run before applying it to all participants in the study. From this initial test run, we learned that, in some cases, the conductivity between the participant and the wristband (Empatica E4) did not gather data in line with what we expected. Some anomalies included little to no detected activity and substantial value differences between participants. Furthermore, we noticed that the activity would vary a lot between the start and end of a gameplay session. The primary cause of this was mostly due to the dry conditions in which the data was collected. For the wristband to accurately pick up a person's sensor data, the electrodes need some sweat to act as a conductor between the skin and wristband. Based on these observations, we changed the protocol to include a short warm-up session before playing the video game and a 15-minute period where the participant would sit still to develop a baseline. The warm-up consisted of walking up and down a flight of stairs spanning six floors two times. This exercise was selected based on tests with some people in the laboratory. The final protocol is shared with the dataset.

Before playing, the participants were told that their performance in the game would be measured based on how many stages were cleared in the time given and on the number of player avatar deaths. Furthermore, we informed participants that their performance would be measured against other participants and that there would be a prize for the highest achiever. The motivation behind making the game more competitive was to make the players want to perform well, and feel like there was some consequence if they either died (in the game) or did not beat levels fast enough. The number of points earned by each player was kept secret from all participants to avoid them giving up or relaxing due to other players too high or too low score. Scores were calculated based on two primary factors; the number of deaths and levels cleared. Starting a level, the player starts with a base score of 1000. For every death, the score is reduced by 100 points down to a minimum of 200. If the player manages to beat the level, he/she is awarded between 200 and 1000. If the player runs out of time, he/she is awarded 0 points and is moved to the next stage. The final score of each participant is included in the dataset and can be seen in Table 1.

After the participants had established a sufficient baseline, they started the primary game session where the video of the participants, video game frames, and sensor data was collected. Each game session lasted for approximately 35 minutes. The game was played directly in the gym-super-mario-bros environment [13], which is a gym [1] based environment for *Super Mario Bros.* For reproducibilities sake, the repository for the gym environment has been added to the official GitHub repository of Toadstool[2]. There are four different graphic environments offered by gym-super-mario-bros, which include the standard graphics, as well as three different downsampled versions (downsample, pixel, and rectangle). The data in our dataset is collected in the standard environment, but sessions may be replayed in any environment if needed.

The gym environment version of the game still differs somewhat from the original gaming experience found in *Super Mario Bros.* by Nintendo (a consumer electronics company from Japan). Firstly, all game-freezing animations and cutscenes are removed from the game. This includes transitions between levels and traveling through pipes. Second, there is no music or other sound effects. Third, there are no limits on game lives, and power-ups do not carry over to new stages. Last, the order of the levels has been changed compared to the original game. There was one additional rule we told participants before playing. In *Super Mario Bros.*, some pipes can warp the player to a new stage that is closer to the final stage of the game. To keep the levels played consistent between players,

[1] https://creativecommons.org/licenses/by-nc/4.0/

[2] https://github.com/simula/toadstool

we asked participants to refrain from using any of the available warp pipes (one located in world 1-2 and two located in world 4-2). Overall, it took approximately one hour to collect data from a single participant.

3 DATASET DETAILS

For each participant, we have included a video of them playing the game (camera facing the face), the controller input performed on each frame of the game, and the sensor data collected from an Empatica E4 wristband [6]. The camera used to collect the facial expression data was a 1.3-MP webcam attached to a Samsung Series 9 Notebook NP900X4C. The webcam captured video at 30 frames per second with a resolution of 640×480. The controller used to play the game was a wired USB controller from retro-bit, which is modeled after the original controller for the Nintendo Entertainment System. Note that the video game frames are not included in the dataset, but can be extracted by using the provided video game actions files included with each participant. This can be done by using a script that is included in the dataset. The reason for not including the video game frames was mostly due to the exponential increase in storage size. Another possible benefit of this approach is the ability to replay the game session in any of the several environments offered by the gym-super-mario-bros framework to produce different representations of game frames. The first frame for each of the 32 levels of *Super Mario Bros.* can be seen in Figure 1. The dataset contains the following files:

- **participants** is the directory that contains the information of each participant. This includes the video of them playing, the controller input of each game frame, and the Empatica E4 wristband sensor data.
- **scripts** is a directory that holds a set of Python scripts meant to aid the user in getting an easy start to using the dataset. The files include a script for replaying gameplay using the provided controller inputs, a script for matching the gameplay session to the facial expression video, and a script for matching the raw signal outputs to the gameplay session.
- **protocol.pdf** is the protocol used to collect the video game session data.
- **questionnaire.pdf** is the questionnaire that was filled out by each participant before starting the game session.
- **questionnaire_answers.csv** is a summary of all the answers to the questionnaire.
- **consent.pdf** is the consent form that was signed by each participant.
- **README.txt** is a short information file which describes the contents of the dataset.
- **LICENSE** is the file that signifies which license in which the dataset is distributed under.

Contained within the *participants* directory is a separate directory per participant included in the dataset. Each directory has a name corresponding to the participant's ID, i.e., *participant_<ID>*, where *<ID>* is replaced with the ID of the participant. For each participant, we have stored the participant's sensor data collected from the Empatica E4 wristband, a JSON file containing information about the participants game session, the video recording of the participant playing the game stored in ".avi" format, and another JSON file which contains information about the video. The JSON file that holds the game session data, called *participant_<ID>_session.json*, contains the actions performed during the game, the start and end times of the game session, and the achieved gameplay score. As for the sensor data, the wristband uses four separate sensors to collect different sensory outputs from the wearer, such as the electrodermal activity (EDA), interbeat intervals (IBI), heart rate (HR), and blood volume pulse (BVP). The four sensors of the wristband is a photoplethysmography sensor, an electrodermal activity sensor, 3-axis accelerometer, and an optical thermometer. Of the four sensors, the thermometer is the only one not graded for clinical use. All data collected by the wristband are stored in CSV files that can be downloaded from the wristband. For the dataset, these CSV files have been matched to the game session. We have also opted to include the raw source files as they were collected from the wristband. The CSV files and a short description of the contents are further explained below.

- **ACC.csv** contains the data collected from the 3-axis accelerometer sensor in the range [-2g, 2g] sampled at 32 Hz. The accelerometer measures the movement of the wearer.
- **EDA.csv** holds the data collected by the EDA sensor sampled at 4 Hz. EDA measures the electrical conductivity of the skin and measurements have been proven to be correlated with emotions since the late 1800s [5]. EDA is also sometimes called psychogalvanic reflex or skin conductance.
- **BVP.csv** contains the data collected from the photoplethysmograph sensor, which measures the BVP, and is sampled at 64 Hz.
- **IBI.csv** stores the interbeat intervals (IBI). The IBI measures the time interval between individual heartbeats and can be used to estimate the instantaneous heart rate as well as heart rate variability. The wristband calculates the values contained within this file based on the BVP signals.
- **HR.csv** contains the average heart rate values, computed in spans of 10 seconds. The heart rate measures the number of times a person's heartbeats per minute. Similar to the IBI, these values are calculated based on the BVP signal.
- **TEMP.csv** holds the information collected by the thermometer, which is the temperature of the person playing the game expressed in degrees Celsius (°C) sampled at 4Hz.
- **info.txt** gives a brief description of all variables collected by the wristband.

In addition to the data collected throughout the study, we also include a set of scripts that aim to make the dataset more accessible. First of all, as previously mentioned, video game frames are not included in the dataset. However, we include the necessary information to extract the frames by using the provided video game inputs to "replay" the game session and collect the video frames directly.

4 PRELIMINARY EXPERIMENTS

We performed a set of preliminary experiments to showcase how the presented dataset can be used to train machine learning algorithms and perform simple predictive modeling. In Section 5, we mention that a possible use case for the dataset is to predict the sensor value of the wristband using the video game frames or facial

	CNN		ZeroR	
ID	MAE	RMSE	MAE	RMSE
0	0.076	0.100	0.075	0.099
1	0.104	0.132	0.103	0.131
2	0.071	0.104	0.075	0.100
3	0.050	0.070	0.050	0.070
4	0.078	0.103	0.069	0.094
5	0.091	0.129	0.094	0.121
6	0.091	0.119	0.090	0.116
7	0.110	0.142	0.109	0.139
8	0.061	0.096	0.060	0.090
9	0.126	0.157	0.109	0.134
All	0.105	0.126	0.090	0.117

Table 2: This table shows the results of all experiments of trying to predict the BVP amplitude using video game frames and facial expressions alone. Note that all experiments were trained over three-fold cross-validation.

expressions as input. In the following experiments, we trained a deep convolutional neural network to predict the BVP amplitudes utilizing a combination of the face data and game frame data. This is similar to what McDuff et al. [17] did when modeling the *emotional* input to their self-driving reinforcement agent.

Before the training step, we needed to prepare the input data, i.e., the video game recording and facial expression video. As the gameplay frames were recorded at 60 frames per second, while the facial expression video was recorded at 30 frames per second, we down-sampled the video game frames by skipping every other frame. After that, the BVP amplitudes had to be extracted from the raw BVP signals and matched to the input frames. The reason for not predicting the raw BVP values is due to the cyclic nature of a beating heart. The BVP has the properties of a sine-like wave that is composed of valleys and peaks which appear in tandem with every heartbeat. We are not interested in the exact value on the signal curve, but the highest point of a cardiac cycle, also known as the systolic peak. This peak-value gives us some information about the emotional state of the human player [20]. To extract the BVP amplitudes, we detect a systolic peak in the given BVP signal and measure its height from the baseline. This peak-value is then repeated until the next detected peak, and-so-forth. The result is a square-like signal in comparison to the sine-like wave that is the BVP. The BVP values were then matched with the input data by taking the average BVP amplitude values over one second (64 measurements in total). The extracted peaks used for the experiments are shared on GitHub together with the code used to produce the baseline experiments[3].

We trained one convolutional neural network (CNN) per participant in addition to one trained on all participants mixed. The purpose of training a model on a single participant is because we generally want to model the emotional response of a single person, not necessarily everyone at once. The model was based on the TensorFlow implementation of ResNet50 [8], where we input two video game frames (first and the last frame of one second) and the facial expression from the last frame of the second corresponding to the video game frames. These three images were grayscaled and stacked channel-wise before being processed by the model. In addition to the CNN-based model, we also calculated the error when using the mean of the response variable in the training for prediction (also sometimes called ZeroR), which should give us some indication about how well our model performs. Since the response is a continuous measurement, we used the metrics mean absolute error (MAE) and root mean squared error (RMSE). Furthermore, we used three-fold cross-validation to not bias the results towards a pre-defined split of the data.

The results of predicting the BVP amplitudes using video game frames and facial expressions can be seen in Table 2. We observe that the trained model sometimes outperforms the baseline and sometimes not, which is an indicator of a challenging task, but not impossible. There is still much room for improvement, where methods recurrent neural networks may be used to increase performance, but this is out of scope for this paper. Furthermore, we want to point out that for classification tasks, such as classifying emotional states based on sensor and video data, one should use standard classification metrics such as precision, recall/sensitivity, and F1-score to determine the quality of the trained model. Optimally, one should also report the true positives, true negatives, false positives, and false negatives so that readers themselves can calculate the metrics manually. The code used to run all experiments is available online in the datasets official GitHub repository (https://github.com/simula/toadstool).

5 POSSIBLE APPLICATIONS OF THE DATASET

We expect that this dataset can be used for many different use cases and scenarios. First of all, we imagine it can be used to detect relationships between the player sentiment and the current gameplay state. This could solely be based on the gameplay frames and collected sensor data, or could also be combined with the player's facial expression for further analysis. The uncovered relationships could be interesting when studying how players get invested in video games, and what typical scenarios contribute to a strong reaction from players. A more specific example could be predicting a person's facial expression based on a given game state or degree of progress in a game or level. This could also be expanded to the sensor data, as one could predict the sensory output of the wristband using the game state and facial expressions as input, like photoplethysmography [21] does with the heart rate, but connected to the current game state the player is in. These two problems could be modeled as either a regression or classification problem, depending on the application.

From a game design perspective, correlations between game progress, game state, the input of players, and the emotions and sentiment of the players could enable a whole new approach to experience-based game design. With the possibility to use this as a method for playtesting, game designers can evaluate when and how their crafted experience is (or is not) invoked in the players. Moreover, games can be built around the concept of self-adapting challenge and difficulty by counter-acting unnecessary frustration, boredom, or annoyance by adapting the game to the player's emotion and sentiment. Last but not least, the correlation between the

[3] https://github.com/simula/toadstool

game, emotions, sentiment, and game engagement, especially flow (the state of being fully immersed in an activity while enjoying it), can be investigated [15, 24]. This would lead to a better understanding of what makes games enjoyable and would impact fields like game and media studies, psychology, game engineering, game design, and serious games / educational games.

Another area where this dataset could be used is in the training of emotionally intelligent machines using reinforcement learning. One way to do this would be by training an algorithm to reproduce the physiological signals based on the game-state. The reproduced signals can then be incorporated into the reward function of a reinforcement learning agent. Since physiological signals like BVP and EDA are reliable indicators of emotional states in humans [12, 19], such an approach could be used to mimic human emotional responses. These kinds of emotionally intelligent machines might aid the pure logic of reinforcement learning models and produce improved learning, as well as reveal new insights into how both machines and humans learn. They might also be a step towards machines with even more complex emotional intelligence, like the ability to recognize, express, and respond to human emotions. This is an active research area leading to better personal assistants, more believable automated communication, as well as more enjoyable and believable video game AI.

Some more concrete examples of possible research questions or experiments are:

- How are the sensor measurements related to the facial expressions?
- Can sentiment analysis of the facial expression be connected to measurements in the sensors?
- How can different data sources be combined efficiently (sensor data with videos, etc.).
- Can game actions of the human players be used as a baseline for human performance?
- Can the additional data collected from people be used to train reinforcement learning algorithms?
- Would a model trained on input from non-experienced players behave differently from a model trained on experienced players?
- Can immersion, enjoyment, and flow automatically be inferred from the gathered data?

As one can see from the discussion above, the dataset holds a lot of interesting research potential to follow up, and we hope that other researchers get inspired to work on the dataset.

6 CONCLUSION

In this paper, we present *Toadstool*, a new dataset consisting of people playing *Super Mario Bros.* and physiological signals corresponding to their emotional reaction to playing said game. While the dataset provides a good starting point for applied research in affective computing in games, emotional AI agents in games, playtesting, and game analytics, we strongly believe the *Toadstool* dataset will foster research in many ways and, therefore, we have detailed follow up research questions in several fields, including affective computing, psychology, and game and media studies. With the detailed description of the dataset, the in-depth discussion of the method included in the dataset, and the wide availability of the game instance used for the experiment and the medical sensors employed, we ensured reproducibility and extension of the dataset. We hope that our work inspires and encourages interdisciplinary and multidisciplinary research to (i) examine how the human element in human-computer interaction can be employed to improve existing machine learning methods by introducing emotional aspects to an otherwise cold and unfeeling machine and (ii) show how interactive entertainment systems, and especially video games, utilize sensory input to provide a personalized and tailored user experience.

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540

[2] Prudhvi Raj Dachapally. 2017. Facial emotion detection using convolutional neural networks and representational autoencoder units. *arXiv preprint arXiv:1706.01509* (2017).

[3] Benedetto De Martino, Dharshan Kumaran, Ben Seymour, and Raymond J Dolan. 2006. Frames, biases, and rational decision-making in the human brain. *Science* 313, 5787 (2006), 684–687.

[4] Liyanage C De Silva, Tsutomu Miyasato, and Ryohei Nakatsu. 1997. Facial emotion recognition using multi-modal information. In *ICICS 1997*, Vol. 1. IEEE, 397–401.

[5] Ch Fere. 1888. Note sur des modifications de la resistance electrique sous l'influence des excitations sensorielles et des motions. *Compt Rend Soc de Biol* 8 (1888), 217–219.

[6] M. Garbarino, M. Lai, D. Bender, R.W. Picard, and S. Tognetti. 2014. Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *ICWMCHM 2014*. 39–42.

[7] Uri Hasson, Samuel A Nastase, and Ariel Goldstein. 2020. Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks. *Neuron* 105, 3 (2020), 416–434.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE CVPR*. 770–778.

[9] Michael Hechter and Satoshi Kanazawa. 1997. Sociological rational choice theory. *Annual review of sociology* 23, 1 (1997), 191–214.

[10] Hong Jiang and Jose M Vidal. 2006. From rational to emotional agents. In *AAAI CMASS 2006*.

[11] Eiman Kanjo, Eman M.G. Younis, and Chee Siang Ang. 2019. Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection. *Information Fusion* 49 (2019), 46 – 56.

[12] Krisztian Kasos, Szabolcs Zimonyi, Eniko Kasos, Avraham Lifshitz, Katalin Varga, and Anna Szekely. 2018. Does the electrodermal system "take sides" when it comes to emotions? *APBF Journal* 43, 3 (2018), 203–210.

[13] Christian Kauten. 2018. Super Mario Bros for OpenAI Gym. https://github.com/Kautenja/gym-super-mario-bros.

[14] Anil Kumar K.M., Kiran B.R., Shreyas B.R., and Sylvester J. Victor. 2015. A Multimodal Approach To Detect User's Emotion. *Procedia Computer Science* 70 (2015), 296 – 303.

[15] Derek A Laffan, John Greaney, Hannah Barton, and Linda K Kaye. 2016. The relationships between the structural video game characteristics, video game engagement and happiness among individuals who play video games. *Computers in Human Behavior* 65 (2016), 544–549.

[16] Jennifer S. Lerner, Ye Li, Piercarlo Valdesolo, and Karim S. Kassam. 2015. Emotion and Decision Making. *Annual Review of Psychology* 66, 1 (2015), 799–823.

[17] Daniel McDuff and Ashish Kapoor. 2019. Visceral Machines: Risk-Aversion in Reinforcement Learning with Intrinsic Physiological Rewards. In *ICLR 2019*.

[18] David Palumbo-Liu. 2005. Rational and Irrational Choices: Form, Affect, and Ethics. *Minor Transnationalism* (2005), 41–72.

[19] Rosalind W. Picard, Szymon Fedor, and Yadid Ayzenberg. 2016. Multiple Arousal Theory and Daily-Life Electrodermal Activity Asymmetry. *Emotion Review* 8, 1 (2016), 62–75.

[20] R. W. Picard, E. Vyzas, and J. Healey. 2001. Toward machine emotional intelligence: analysis of affective physiological state. *IEEE Transactions on PAMI* 23, 10 (Oct 2001), 1175–1191.

[21] Philipp V Rouast, Marc TP Adam, Raymond Chiong, David Cornforth, and Ewa Lux. 2018. Remote heart rate measurement using low-cost RGB face video: a technical literature review. *Frontiers of Computer Science* 12, 5 (2018), 858–872.

[22] John Scott. 2000. Rational choice theory. *Understanding contemporary society: Theories of the present* 129 (2000), 671–85.

[23] Shiv Naresh Shivhare and Saritha Khethawat. 2012. Emotion detection from text. *arXiv preprint arXiv:1205.4944* (2012).

[24] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)* 3, 3 (2005), 3–3.

Questionnaire for Data Collection Study

Super Mario Bros Data Collection Study

Please answer these questions to help us analyze the gathered information correctly.

1. What is your age?

_____

2. What is your sex?

Mark only one oval.

- ◯ Male
- ◯ Female
- ◯ Rather not say

3. Are you left-handed or right hand-handed?

Mark only one oval.

- ◯ Right-handed
- ◯ Left-handed
- ◯ Neither

4. How many years have you played video games actively (at least once a week) in the past?

_____

5. How many hours do you currently spend playing video games a week?

Mark only one oval.

- ◯ 0-1 hours
- ◯ 1-4 hours
- ◯ 4-8 hours
- ◯ 8+ hours

6. How much prior experience do you have playing Super Mario Bros or other similar platforming games?

Mark only one oval.

- ◯ I never played them
- ◯ Have played such games on occasions but never consistsently
- ◯ I have played most of or finished a few (1-2) such games
- ◯ I have played and finished several (3+) such games

7. Do you consider yourself to get easily exited, stressed or otherwise emotionally aroused in your daily life?

Mark only one oval.

	1	2	3	4	5	
No, it takes a lot	◯	◯	◯	◯	◯	Yes, very easily

Consent Form for Data Collection Study

Consent form

In this data collection study you will be playing Super Mario in a specially set up environment. As you play we will collect information through the game, a computer camera and a wrist sensor (Empathica 4). The main purpose of the study is to collect data that may be used to train machine learning algorithms on human signals to be able to synthesize such signals (creating similar but not real responses).

For this study we will record three types of information from you;

1. The frames, as well as the actions performed, from the game environment
2. Physiological sensor data including electrodermal activity, heart rate, blood volume pulse and skin temperature.
3. Video recordings of your face as you play. The video recordings do not include audio.

The recorded data will be used for emotional and sentiment analysis to train an artificial agent to learn to play a game. No names or other personal or identifying information will be recorded beyond the video images, and the data will be kept as anonymous as possible.

Do you consent to have your in-game actions, as well as your physiological data collected from sensors, recorded, shared and published for research purposes?

- ❑ Yes
- ❑ No

Do you consent to have a video of your face while playing the game recorded, shared and published for research purposes?

- ❑ Yes
- ❑ No

Signature *Date*

Protocol for Data Collection Study

Protocol for Data Collection on Super Mario Bros

Goal of the study:

The purpose of the study is to gather data to be used in a reinforcement learning framework for an agent to learn to play Super Mario Bros. We will collect sensory information on the physical and mental states of human participants that maps to states and actions in the game environment. This data will be used to train one or more generative neural network algorithms to generate signals that imitate human reactions to game states and correlates to game situations that could be relevant for learning. The idea is to include such signals in the reward function of a reinforcement learning agent to observe how this might affect the learning process.

Participants:

We aim to have a minimum of 10 participants with a balanced distribution of genders and gaming experience.

Collected Data:

Sensor data:
We will collect data through the empatica 4 which allows real time high quality readings of;

- BVP (Blood Volume Pulse)
- IBI (Inter Beat Interval)
- EDM (Electrodermal Activity)
- XYZ raw acceleration
- Skin temperature

Video data:

- Real-time raw video of participants as they play
- Sentiment analysis/Emotion in face

Game data:

- Game state (pixel image)
- Actions pressed on the keyboard

Game-Environment:

Participants will play in the gym-super-mario-bros environment which runs in the OpenAI-Gym framework. This is a framework made for reinforcement learning agents so it has some key differences from the original game environment. Including;

- Game-freezing animations and cutscenes have been removed
- There is no music or sound-effects
- There is no limit on lives
- Power-ups will not carry over to new stages
- Warp transitions to new stages will not count as finishing a stage

Setup:

- Participants are asked to play 35 minutes
- During the first 5 minutes participants may be instructed on certain game mechanics and strategies
- The participants will be told that it is a competition and they have to try to complete as many stages as they can while dying as few times as possible.
- Participants will be observed as they play by a researcher
- The stages will be given in predetermined order slightly different from the original game

Consent form:

Participants will be asked to sign a consent form, confirming their consent to recording, sharing and publishing of;

1. Their in-game actions and physiological data collected from sensors
2. Video of their face while playing the game

Questionnaire includes:

- Age
- Sex
- Dominant hand
- Gaming experience
- Mario experience
- Level of anxiousness/nerves/excitement.

Equipment:

1. Gaming setup
    a. Laptop
    b. Play in the OpenAI gym environment
    c. Frame and key-logger
    d. Gamepad
2. Empatica E4 watch and data collector (management suite)
3. Video camera for facial affective analysis (laptop camera)
4. Storage for data

Study protocol:

Step by step walkthrough of a data collection session.

1) Attach E4
2) Sign consent form
3) Fill in questionnaire
4) Warm-up to build sweat for EDA sensors - 5-10 min of light exercise like walking stairs
5) Establish baseline for sensors - 15 min
6) Get task explained
7) Let participant play
    a) Collect data during game
        i) Sensor data from E4
        ii) Game data (actions, frames of the game)
        iii) Facial expression or sentiment of the facial expression
    b) Play rules
        i) Unlimited lives
        ii) 5 extra minutes to get used to the controls and game
        iii) 30 minutes playtime
        iv) The first five levels represent the five different environments:
            (1) Overworld
            (2) Underground
            (3) Athletic
            (4) Castle
            (5) Underwater
        v) Level 6 and 7 contain Hammer Bro’s and Lakitu
        vi) After that all unplayed stages in sequential order
        vii) If a participant dies after about 5 minutes on one level we skip to next level
        viii) A score is calculated based on the number of stages completed and number of deaths;
            (1) 1000 points base score for completing a level
            (2) For each death on a level base score is reduced by 100 down to a minimum of 200
            (3) If the player is transported to the next level due to timing out, no points are received
8) After the session is completed and recorded, the E4 is removed.