Affective and Human-Like Virtual Agents

by

Neil Bhavendra Budnarain

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In Artificial Intelligence (AI) one of the technological goals is to build intelligent systems that not only perform human level tasks efficiently, but can also simulate and exhibit human-like behaviour. As the emphasis of systems is often placed on fulfilling functional requirements, AI systems are only intelligent at a machine level. Affective computing addresses this by developing AI that can recognize, understand and express emotion. In this work, we study the effects and humanness of emotionally cognizant AI agents within the context of the prisoner's dilemma. We leverage machine learning techniques and deep learning models in devising algorithms to map dimensional models of emotion to facial expressions for virtual human displays. Additionally, we utilize distributed representations for words to design a method for constructing affective utterances for a virtual agent in the prisoner's dilemma. We experimentally demonstrate that our methods for affective facial expression and utterance construction can be successfully used in AI applications with virtual humans. Thus, we design and build a prisoner's dilemma game application including the integration of a virtual human. We conduct two experiments to study and evaluate humanness of various agents in the prisoner's dilemma game. We demonstrate the effectiveness of our facial expression and utterance methods and show that an appraisal-based theoretic agent is perceived to be more human-like than baseline models.

Acknowledgements

I would first like to thank my supervisor, Professor Jesse Hoey. It's very rare to find a professor with such humility and sincerity. Jesse gave me the freedom to pursue research topics of my choosing. His creative intuition and ability to cleanly formalize complex problems is truly inspiring and has impacted the way in which I think. I could have not asked for a better supervisor. I owe a debt of gratitude to him for his guidance, support, and kindness.

I would like to thank Professor Pascal Poupart for providing incredible mentorship during my undergraduate studies, and for supporting my pursuit of graduate studies. To my committee members: Professor Edith Law and Professor Christopher Batty. Thanks for taking the time to read and offering feedback of my thesis.

I would also like to thank all the members of our research group at the University of Waterloo. I would like to acknowledge Dr. Moojan Ghafurian. A special thank you to Joshua Jung. Josh has been incredibly helpful throughout my masters, especially during my early beginnings at the lab. I would also like to thank Aarti Malhotra for her help, and collaboration over my research.

I would like to thank Kudas for his constant encouragement and support during my studies. I want to thank my cousin Ovendra for his support and for helping me move many times to and from Waterloo. I would also like to thank my uncle. Uncle Andre has been instrumental in sparking my interest in computer science and I am very thankful for his invaluable advice in traversing life.

I would like to thank my mom for her unconditional love and support. I am forever grateful for my mom. My mom has made tremendous sacrifices in support of my education. It goes without saying that this achievement and degree is much more hers, than it is mine.

Although my dad did not live to see this day, I am truly thankful and deeply grateful for him. He has always been a source of motivation during my studies and this would not have been possible without him.

Thanks to my brother for helping me move many times to and from Waterloo. We have continuously pushed each other intellectually and together have always found a way in making what may seem unattainable attainable.

Finally, I would like to thank my girlfriend Stacy. Thank you for your love, support, and for being the amazing person that you are.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

A key goal of Artificial Intelligence (AI) is to build intelligent systems that can perform human level tasks both efficiently and effectively. The research, design and development that goes into AI aims to build machines that can exhibit intelligent behaviour. AI essentially attempts to mimic or simulate human cognitive functions such as problem solving and learning [67]. Subareas in AI such as machine learning have had tremendous success over the recent decade. Recent breakthroughs in machine learning have resulted in technological advances of various machine learning applications in natural language processing and computer vision. For example, this includes speech recognition systems, facial recognition systems, machine translation tools, conversational agents and smart home systems.

AI-based dialogue systems have gained tremendous traction in recent years. It should be noted that dialogue systems are also referred to as conversational agents or chatbots. From an alternative perspective, dialogue systems can be viewed as intelligent virtual assistive technology [14]. AI speech-based conversational agents have emerged in forms such as Google's Assistant, Apple's Siri, Microsoft's Cortana and Amazon's Alexa. In addition, there are now a plethora of text-based conversational agents deployed in the form of chatbots. Many of these technologies are changing the way in which we as humans interact with machines. Many companies are leveraging this technology and transforming the way they conduct their business. For example, customer service is massively being revolutionized as consumers are using chat services to interact with businesses [3]. This is especially prevalent in e-commerce settings where goods and services are being offered and exchanged through the internet. With constant technological advances in AI, human

chat service operators will continue to be replaced with artificially intelligent conversational agents [3, 27]. In addition to conversational agents, there has been extensive work regarding the integration of virtual human agents into the interface of several AI systems. This has resulted in AI systems having the ability to display both verbal and non-verbal emotional signals. A few examples includes tutoring systems, smart home systems, and assistive handwashing systems to help those with dementia [91, 64, 11, 100].

One significant feature among the use cases listed above, is that they all share one commonality. That is, the human interaction aspect or human to machine interaction. One of the shortcomings in the development of AI systems is that many systems are geared towards meeting more functional requirements, and as a result are often intelligent only at a machine level. As the purpose of a majority of AI systems or agents are often to be deployed in social environments or settings, this often requires a human interaction component. As a result, more significant considerations should be made pertaining to the emotional or affective aspects of the overall system. Therefore, not only do AI systems need to be intelligent at a machine level, they need to be intelligent at a human level.

1.2 Affective Computing

Affective computing is an interdisciplinary field that aims to design intelligently and emotionally cognizant inclined systems that can recognize, feel, interpret, process and simulate human emotion [77, 95]. Emotions and sentiments are a fundamental part of human interaction and the human experience. Emotions and sentiments are ubiquitous in our daily lives and have a significant influence in cognition, our perception, the way in which we communicate, and our decision making. In the majority of technological innovations of AI systems, a significant problem is that these systems lack the ability to emotionally align with human user. This problem is addressed through the field of affective computing. Rosalind Picard, founder and director of the Affective Computing Research Group at the Massachusetts Institute of Technology (MIT) Media Lab, explains that if we want computers and machines to be genuinely intelligent and to interact naturally with and amongst humans, computers must have the ability to recognize, understand and express emotion [75]. The ability of having AI systems that can affectively align with humans in interactions while making intelligent decisions, and being placed complex social environments, is paramount.

In order to design and develop AI systems that are able to effectively understand emotion, and affectively align in human interaction, one of the first steps we have taken was to study emotion intelligence within the context of social dilemmas. Through the usage

of artificial virtual humans, we were able to simulate emotion through verbal and nonverbal cues and study their effects in human to machine interaction. In this thesis we designed and implemented a prisoner's dilemma game with the necessary architectural components enabling the successful integration of a virtual human into the gaming interface. The advantage of integrating a virtual human into the gaming interface has allowed for the usage of verbal and non-verbal cues such as affective facial expressions and affective utterances. As a result, in this work we have studied, designed and generated affective facial expressions and affective utterances mappings by leveraging dimensional emotion models.

1.3 Contributions

The main contributions of this thesis are as follows:

1. Designed and implemented a prisoner's dilemma web-based game with capabilities making of being deployable to conduct online experiments with human participants through Amazon's Mechanical Turk platform. Our robust system is used to study the effects of human emotion and decision making within the context of the prisoner's dilemma. This has led to the study and execution of additional experiments. Our system is a platform that sets the foundation for building, experimenting, and studying human emotion within the context of other social dilemmas.

2. Generated a dataset containing 9,200 virtual human faces and their corresponding configurations. This dataset can be leveraged by machine learning models in learning virtual human facial displays.

3. Experimented with various machine learning and specifically deep learning algorithms in order to generate affective virtual human configuration based on dimensional emotion models. Compared and evaluated the performance of state-of-the-art deep CNN models in the application of dimensional affect prediction.

4. Created a systematic mechanism for mapping affective representations to utterances at the sentence level. These utterances would be communicated verbally by a virtual human. Part of this work entailed leveraging natural language processing techniques such as word embeddings.

1.4 Thesis Organization

This thesis is structured and organized as follows:

- Chapter 2 discusses background and related works.
- Chapter 3 discusses the prisoner dilemma system architecture and overall system design used in our experimental research.
- Chapter 4 discusses the affective facial expressions mappings based on dimensional emotion models from the continuous domain.
- Chapter 5 discusses the affective utterance mappings based on dimensional emotion models from the continuous domain.
- Chapter 6 discusses the prisoner's dilemma experiment and results that we ran with human participants.
- Chapter 7 contains the conclusion and some further discussion regarding future work.

Chapter 2

Related Work

2.1 Social Dilemmas

A social dilemma is a social situation where when one focuses on themselves it ultimately leads to everyone else being worse off. Therefore, a social dilemma may be described or defined as "a situation in which individual rationality leads to collective irrationality" [47]. These social situations are present everywhere and are often at times the root of many non trivial problems we face in the world. For instance, something as simple as jumping a line at the grocery store, to something more complex such as social traps can be considered social dilemmas. Social traps occur when immediate rewards are pursued and are later revealed to be "unpleasant" or in some cases "lethal" [76]. Overfishing, the impact that automotive vehicles have on air pollution, and deforestation, are all examples of social traps where at their core are social dilemma constructs. The study of social dilemmas has sparked the development of modelling approaches and solutions which has in turn provided extensive and widely used applications.

These social situations are important because they can be leveraged to study the effects of the interaction of artificial agents exhibiting human emotion and human behaviour. While social dilemmas can be modelled in a variety of ways we focus on one specific model which is categorized as two-person dilemmas.

2.1.1 The Prisoner's Dilemma

In 1950, two scientists at RAND corporation, Merrill Flood and Melvin Dresher, created what is known to be the simplest example of a two person game which they later ran as an

informal experiment in their research [47]. This game later became known as the prisoner's dilemma and came with a backstory both of which created by Albert W. Tucker [62]. The prisoner's dilemma is a social dilemma which involves two people who have the choice between two options. One option typically representing cooperation and the second option representing defection. While there are many depictions of the backstory, [62] describes the story for this game as follows.

The main story line starts off with the premise that the police have charged two men who have jointly violated the law and are both being held separately by the police. They have separately been given a choice between two options: They can either confess. Or they can refuse to confess and keep quiet. Since both prisoners are separately given this option this is where things get interesting. In this conceptualization of the prisoner's dilemma, confessing can be observed as defection. There are 4 possible outcomes in the given scenario.

From [99] consider the following:

1. If both men confesses then each will be charged with 1 unit.
2. If one confesses and the other does not confess then the one who confesses will be rewarded with 1 unit and the latter charged with 2 units.
3. If both men refuse to confess then both will be free and will go clear.

In game theory, the prisoner's dilemma is considered a non-zero sum game [62]. That is, games where one's gain or loss are not balanced by the opposing participant's gain or loss. In a zero sum game a participants gain or loss is balanced by the other participants gain or loss [80]. For example, many gambling games such as poker are considered a zero sum games as one's gain is the combined loss of others.

Many generalizations of the prisoner's dilemma have been made stemming from Tucker's initial interpretation. One common example from [47] of the prisoner's dilemma utilizes the following generalization. Two students are asked to take out $1 from their wallet and both separately in private are given the option of either putting their dollar into an envelope or keeping their dollar. The act of keeping their money can be considered defection and the act of putting their money into the envelope can be considered cooperation. The envelopes are swapped and the money that is given in the envelope is doubled by the teacher, thus resulting in an additional reward for cooperation.

We summarize the payout matrix given this generalization of the prisoner's dilemma in Table 2.1. All of the possible outcomes are shown in the payout matrix in Table 2.1.

While the socially optimal or globally optimal solution would be to have both students cooperate, the Nash equilibrium is the state at which both students defect. This may seem counter intuitive but when factoring in what the other player can potentially do, you would observe that both players defecting can be considered a deficient equilibrium. For example, if player I cooperated and player II defected then player II would be rewarded with $3 while player I collects nothing in return. Therefore, given the uncertainty of each player not knowing each other's move it would seem best and safest to defect, hence why defection is considered the Nash equilibrium.

Payoff Matrix		
	Cooperation	Defection
Cooperation	**2**,2	**0**,3
Defection	**3**,0	**1**,1 *

Table 2.1: Payout matrix for prisoner's dilemma game.Two person prisoner's dilemma payout matrix taken from: player I and player II where player I's payout in bold and the asterisk denotes the Nash equilibrium [47].

The following inequality describes the ranking of the moves by maximizing the payout to player I:

$$DC > CC > DD > CD \tag{2.1}$$

For example, from Equation 2.1, DC would indicate that Player I defected as shown by D, and Player II cooperated as shown by C.

2.1.2 Other Two-Person Social Dilemmas

In addition to the prisoner's dilemma, there are also two other two-person social dilemmas. The first is called the assurance game.

$$CC > DC > DD > CD \tag{2.2}$$

Adjusting this inequality or in other words moving the relative value of these outcomes may result in other social dilemmas. If the inequality is modified where mutual cooperation is the most rewarding outcome, this reflects the social dilemma known as the assurance

game. The inequality representing the assurance game is shown in Equation 2.2. In the assurance game mutual cooperation is considered the optimal equilibrium while mutual defection is the deficient equilibrium. The game is premised on the issue whether each player will trust each other in terms of cooperation [61, 47]. The following is an example of the typical scenario in which the assurance game stems from. The scenario states that we have two hunters have a dilemma of either hunting a stag or a hare. In order to successfully capture the stag this would require mutual cooperation amongst the hunters. Since the stag is a much larger animal, capturing it is considered a more fulfilling meal [61].

$$DC > CC > CD > DD \tag{2.3}$$

If the inequality is adjusted where mutual defection results in the worst outcome, then this is known as the chicken game. The rankings of the outcomes of the chicken game are shown in Equation 2.3. The idea or scenario of the chicken game is as follows: There are two individuals driving their car into each other. The first one to deviate and turn away will be the "chicken". The individual who does not turn away first will be the winner or in other words seen as more courageous or brave. If neither individual drives away then they both will collide into each other resulting in both of them dying, hence the worst outcome. There are another class of social dilemmas that deal with more than two players referred to as multi-person dilemmas [47]. However, in this work we focus on two person dilemmas and in particular, the prisoner's dilemma.

2.2 Appraisal Theory

The process of emotion elicitation is the premise of appraisal theories [86]. Rather than placing emphasis on the consequences or precursors of an emotional reaction, appraisal theory focuses on distinguishing and reasoning about emotional states based on the evaluation of eliciting conditions [58]. Examples of these eliciting conditions may include aspects such as one's inherent pleasantness, or one's goals [58]. This means emotions are evoked by appraisals of situations or events. An example of the emotion of sadness being felt when one loses a loved one may be elicited by the evaluation/appraisals where someone that was once around, has been lost and now gone. Malatesta et al. explains that appraisal theories are a common approach when it comes to emotion modelling [58]. This is due to the fact that the underlying structure of appraisal theory makes it practical to simulate in computational models and implementing in computers [58]. There are many popular theories in this area such as Scherer's appraisal theory [85], Roseman's theory [81], Frijda's theory[25],

and the OCC theory[68]. In this work we focus on the OCC theory as we implemented and studied the OCC model in our experiments with human participants.

2.2.1 OCC Model

The theory of Ortony, Clore and Collins (OCC) is considered one of the most successful models used to classify emotion [95, 68]. In comparison to other appraisal theories, OCC has been shown to best fit in the implementation of virtual agents [2]. Its simplicity and the finite set of appraisal combinations that it consists of, make it a common choice by computer scientists in building AI agents that are programmed to model and reason about human emotion [2].

The OCC model is based on cognitive elicitors and as a result, is considered a cognitive appraisal theory [2, 58]. The OCC model aims to classify people's emotion based on objects, events, and agents. Objects, events, and agents make up a three branch typology which refers to three types of stimuli components of the model. These are the aspect of objects, actions of agents, and consequences of events. [2]. In other words, this means for people, if happiness or unhappiness is associated to an event, there is either a like or dislike of an object, and approval or disapproval of an agent [95]. Consequently, people's emotions are categorized based on this three branch typology/categories. Under these three categories there are a total of 22 described emotions or emotion types.

The OCC model contains 22 emotions that are grouped into six classes: Fortunes of others, prospect-based, well-being/attribution compounds, well-being, attribution, and attraction. For example, when a person is appraising an event that has occurred, well-being emerges when the desirability of the consequences are all for him or herself resulting in emotions such as joy and distress. Fortunes of others emotions emerge when one appraises an event, but the emphasis is on the desirability on the other individual resulting in happy-for, resentment, gloating, and pity. With similar reasoning the rest of the branches of the OCC model can be explored [68].

2.3 Dimensional Models of Emotion

In addition to categorical models of emotions which are discrete and categorical, there exists dimensional models of emotions. Dimensional models of emotion have been utilized in conceptualizing human emotion in multiple dimensions, often two or three dimensions. This means that emotions are mapped into two or more higher dimensional space. One of the

most commonly used dimensional models is the circumplex model. This has been proposed by Russell, and maps emotions into a two-dimensional space where each dimension is represented by valence and arousal respectively [83]. Valence measures pleasantness versus unpleasantness and arousal measures active versus inactive or in other words the emotion activation. Later work has resulted in the addition of a third dimension, dominance [49]. Valence, dominance and arousal are congruent with evaluation/potency/activity (EPA) space. These dimensional models of emotion have made it possible to study more complex relationships in facial expressions, sentiment analysis captured in valence, arousal, and dominance dimensions.

2.4 Affect Control Theory

A sociology theory called Affect Control Theory (ACT) has been developed from language-based mathematical models of impression formation, emotion and attribution [87]. ACT claims that we all as individuals carry a fundamental feeling or sentiment about who we are in society. In other words, these fundamental sentiments can be considered to be an emotional feeling about ourselves and are used to represent social identities, and behaviours. Given these fundamental sentiments, when we as individuals engage in social interactions or situations, transient impressions are formed as a result of an event. ACT further proposes that whether it be emotion, social behaviour, or social interaction, there is psychological need to minimize the difference in the fundamental sentiments and transient impressions. This difference is referred to as the deflection. As a result, the main principle of ACT is that actors work to have transient impressions that are inline or consistent to the fundamental sentiments of the actor. ACT describes events occurring in the world or society through a grammatical structure referred to as an actor-behaviour-object model [36]. An actor refers to an entity who chooses to initiate some sort of event. A behaviour is specified by a verb describing something that the actor does. An object is the receiver of the actor's behaviour. The objects, are themselves also actors.

Three-dimensional vectors are used to represent or describe emotions and sentiments. These three basis vectors form the affective vector space known as Evaluation (E), Potency (P), and Activity (A). Evaluation is used to measure how pleasant or unpleasant something is. Potency is used to measure how powerful vs powerless something is. Activity is used to measure how exciting or calm something is. EPA profiles are used to measure concepts through a semantic differential scale [70, 69]. Semantic differential scales are numerical bipolar scales where opposite adjectives are placed at the ends and utilized in rating concepts affectively. The convention has been formed to utilize scales ranging from

-4.3 to +4.3. For example, when measuring the evaluation dimension, the use of a semantic differential results in a numerical scale where the closer the rated concept is to +4.3, the more good/pleasant the concept is, and the closer the rated concept is to -4.3, the more bad/unpleasant. Therefore such a numerical scale is able to capture intensity indicating the significance or strength of the corresponding adjective used to describe the concept. It has been shown that people from similar cultural backgrounds share consistent agreement in EPA ratings in measuring concepts. Sociologists have surveyed human participants from similar cultural backgrounds and have built affective lexicons of concepts, behaviours and emotions rated on evaluation, potency, activity scales. For example, a suspect is represented in EPA form as [-0.87,-0.34,-0.08], a student [1.49,0.31,0.75], and the emotion of happy [2.92,2.43,1.96][1].

ACT has a mathematical formation [36] and is briefly described as follows:

The fundamental sentiments $\mathbf{f} \in [-4.3, 4.3]^9$, are represented by a 9 dimension vector containing three sets of EPA values, each of which correspond to the actor, behaviour and object of the model. Transient impressions $\boldsymbol{\tau} \in [-4.3, 4.3]^9$, are represented in a similar manner. The deflection is the difference between the fundamental sentiment and the transient impression, measured by 2.4, which is Euclidean distance where w_i represents summation weights.

$$D = \sum_i w_i (f_i - \tau_i)^2 \tag{2.4}$$

In ACT, there is a mathematical predictor function that enables users to predict the transients from the fundamentals through regression with a set of non-linear features and is modelled by a formula shown in 2.5.

$$T_{t+1} = \boldsymbol{M}\mathscr{G}(f, \tau) \tag{2.5}$$

$\boldsymbol{M}$ is a matrix containing prediction coefficients that have been estimated from impression-formation research, and $\mathscr{G}$ is a vector containing pre-event transients and interaction terms that have been shown to be relevant through empirical analysis. The emotion that is felt resulting from an event is proportional to the vector difference of the fundamentals and the transients as shown in Equation 2.6.

$$e \propto (f - \tau) \tag{2.6}$$

[1] All of these EPA values are taken from the Indiana 2002-2004 dataset [37].

In ACT, emotion is essentially computed as a function of the difference of the fundamentals and transients.

2.5 BayesACT

BayesACT is a partially observable Markov decision process (POMDP) [4] model of affective interactions [39]. BayesACT builds on ACT and goes further in formulating a probabilistic and decision theoretic model which is to be considered a generalization of ACT. While BayesACT maintains the same underlying ACT principles, what differentiates BayesACT from ACT, is that it keeps multiple hypotheses about behaviours and identities simultaneously as a probability distribution. The ability of learning about people's identities and the predictive nature of people's behaviour is what allows BayesACT to engage in affectively believable human interaction. It has been shown and demonstrated that BayesACT is effectively able to be integrated into human interactive systems as an emotional plug-in. BayesACT has been integrated into the COACH system. The COACH system is a smart home system which leverages AI to build assistive technology for older people with dementia, and is focused on activities of daily living (ADL) through the usage of using audio/visual prompts [63]. A target ADL such as handwashing has been used in the COACH system. The integration of BayesACT into a handwashing system has been successfully demonstrated in its use as a emotional plug-in for AI human interactive systems [52].

Chapter 3

Prisoner's Dilemma System Architecture

In this work we study the effects of emotion conveyed through a virtual human agent. We have designed and implemented a system that is built based on a generalization of the prisoner's dilemma. The idea is to have a human player play a game against an emotionally intelligent artificial agent, where the artificial agent has capabilities to display or convey emotional signals through speech and facial expression cues. The artificial agent will be displayed in the form of a virtual human through a gaming interface. This system is designed to act as a robust platform for future use in research on additional social dilemma scenarios.

First, we discuss in detail and provide a game description of how the prisoner's dilemma game is constructed within our gaming system environment. After, we will go through an example of a series of interactions between the artificial agent and human player in an effort to demonstrate gameplay in the prisoner's dilemma. This chapter then concludes with a description and details of the individual components that our prisoner's dilemma system architecture consists of.

3.1 Iterated Prisoner's Dilemma Game

3.1.1 Game Description

As discussed in Chapter 2, the prisoner's dilemma is a common example in game theory. For our purposes, we focus on the iterated prisoner's dilemma used in our experimental research. This will be explained shortly.

The iterated prisoner's dilemma game consists of two players. The first being the artificial agent and the second being the human user. The artificial agent and human user will then play a game against each other. The game begins with 4 coins being taken from the pot of coins and then being placed in the middle of the table. At this point, each player is given the choice of making two moves. That is, the human player is given the choice of either cooperating or defecting and the artificial agent is also given the same two choices. Cooperating means giving two of the four coins in the middle to his/her opponent. Alternatively, defecting means taking one of the four coins in the middle to his/her self.

In our gaming interface the human user selects their move by clicking one of the buttons as shown in Figure 3.2 where cooperation is equivalent to "Give 2" and defection is equivalent to "Take 1". "Give 2" represents the action of giving two coins and "Take 1" represents the action of taking one coin. In addition to the choice of move, the human player also selects an emoji expressing their emotion in the given moment as shown in Figure 3.1.

Figure 3.1: Game interface displaying the emojis while hovering over "joy" used for emotion selection after "give 2" was selected

The human player's move and the artificial agent's move are kept hidden and are only revealed to each other after both the human user and the artificial agent have completed making their choices i.e. at the end of the round. The human player's move and the artificial agent's move are illustrated being kept hidden by having a red barrier that blocks the buttons as shown in Figure 3.2.

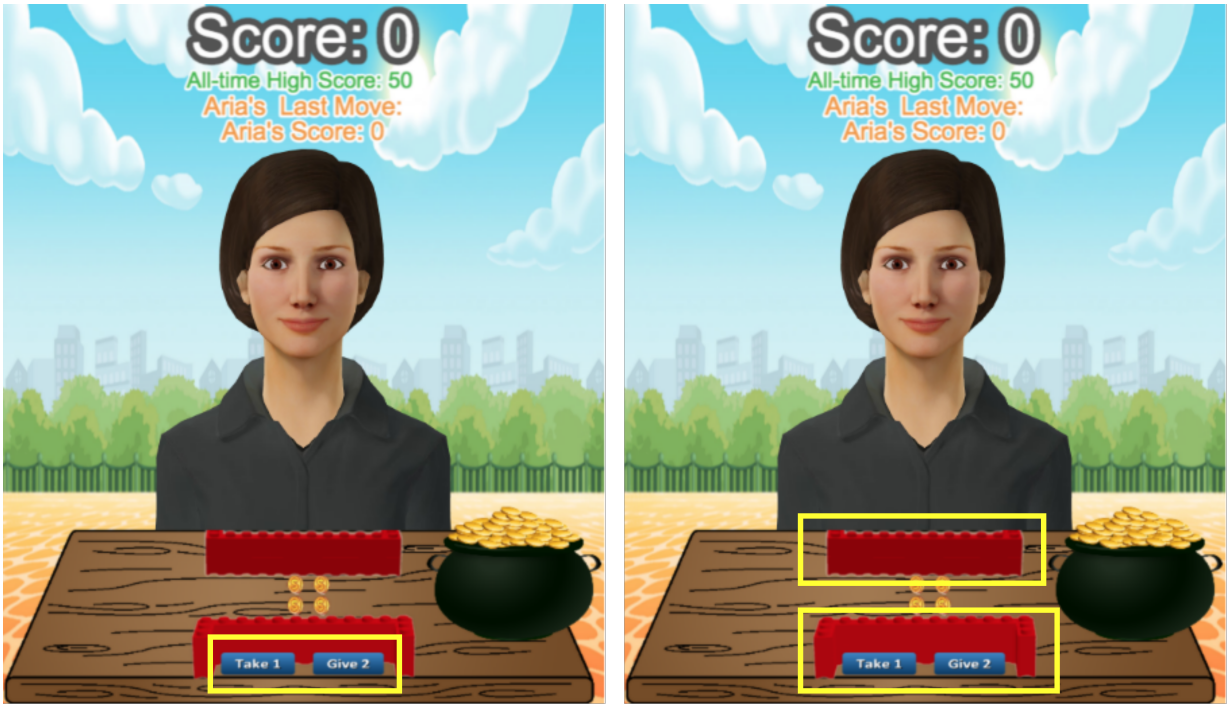


Figure 3.2: Game interface displaying the two buttons for move selection on left and red barrier on the right. "Give 2" button, "Take 1" button and red barriers are highlighted.

After the players choices are made, they are revealed to each other and the 4 coins are distributed to each players respective piles. As the coins are being distributed the virtual human will display two emotional signals in the form of an utterance and facial expression which is meant to reveal or convey the agent's emotion or more generally how the agent is feeling in that given moment.

Once the coins are distributed to each players respective piles, the scores will be updated accordingly based on the number of coins that have been collected. The updated score can be seen at the top of the gaming interface (see Figure 3.2). After the scores are updated this indicates that one round of the prisoner's dilemma has been completed. After the completion of the round, four coins will again be taken from the black pot of coins and placed in the middle indicating the start of another round of the game. Many rounds are then played, hence why the game is referred to as the iterated prisoner's dilemma.

The objective of the game is to maximize your score which means collecting as much gold coins as you can. The payoff matrix displayed in Table 2.1, indicates how the score is updated. For example, if the artificial agent cooperates and the human player defects,

this results in the human player collecting three coins in total and the artificial agent will collect zero coins. As a result, the human player's score will be increased by three points and the artificial agent's score will remain the same. As another example, consider if both the human player and the artificial agent cooperates, this results in both players receiving two points and increasing both their scores by two.

3.1.2 Gameplay Interaction Example

We will now go through an example of the interactions that take place in a single round of gameplay in the iterated prisoner's dilemma.

(a) start of round

(b) human player clicked "give 2" button

(c) human player clicked "happy" emoji

(d) virtual human reacts with a "scared" facial expression. Coins are distributed and scores are updated

Figure 3.3: Interaction of a single round of the iterated prisoner's dilemma

Figure 3.3 shows an overview of a single round of Gameplay in the Iterated Prisoner's dilemma. At the start of the round the "Give 2" and "Take 1" buttons are displayed to the user, as shown in Figure 3.3a. This indicates to the user that he/she has to make their move. This action is made by clicking one of the two buttons. Once the user clicks one of the buttons, it will be shown as selected by turning red and being unclickable. Immediately following move selection, emojis are displayed, as shown in Figure 3.3b. In this example, it is shown in Figure 3.3b that the user has chosen to cooperate by clicking the "Give 2" button. In this example, the user feels happy and specifies their emotional signal by clicking the "happy" emoji representing how they feel at the given moment. Figure 3.3c shows that the user has selected the "happy" emoji. After the user has made their emoji selection the coins are distributed to each players respective piles and the scores are then updated. After the scores are updated, 4 coins are again taken from the black pot of gold coins and are placed in the middle and the "Give 2" and "Take 1" buttons are visible on the interface indicating the start of the next round.

3.2 System Overview and Design

In this section we present details of the overall system design and implementation.

Our main objective is to design and develop a system or platform for studying the effectiveness of emotionally cognizant artificial agents in human interaction within the context of the iterated prisoner's dilemma. Additionally, it is our intention that we have designed and implemented a system that is able to accommodate and generalize well for future research of affective agents applied to various other social dilemma scenarios.

Our prisoner's dilemma game is a web based application that follows a client-server web architecture model. In this client-server model the two primary components are the front-end which is served on the client side and the back-end containing server side code written in Python. The system overview is visualized and displayed in Figure 3.4. The front-end consists of all of the HTML5, CSS, JavaScript code and as well as the Phaser game code which runs on the client side. Two significant components that we have implemented in the front-end is the client side code that integrates the virtual human and leverages the Phaser game framework. As shown in Figure 3.4 we make use of virtual human API and Phaser.io API.

In the following subsections we present a brief description of these components. This includes the back-end, virtual human integration, and the Phaser game framework.

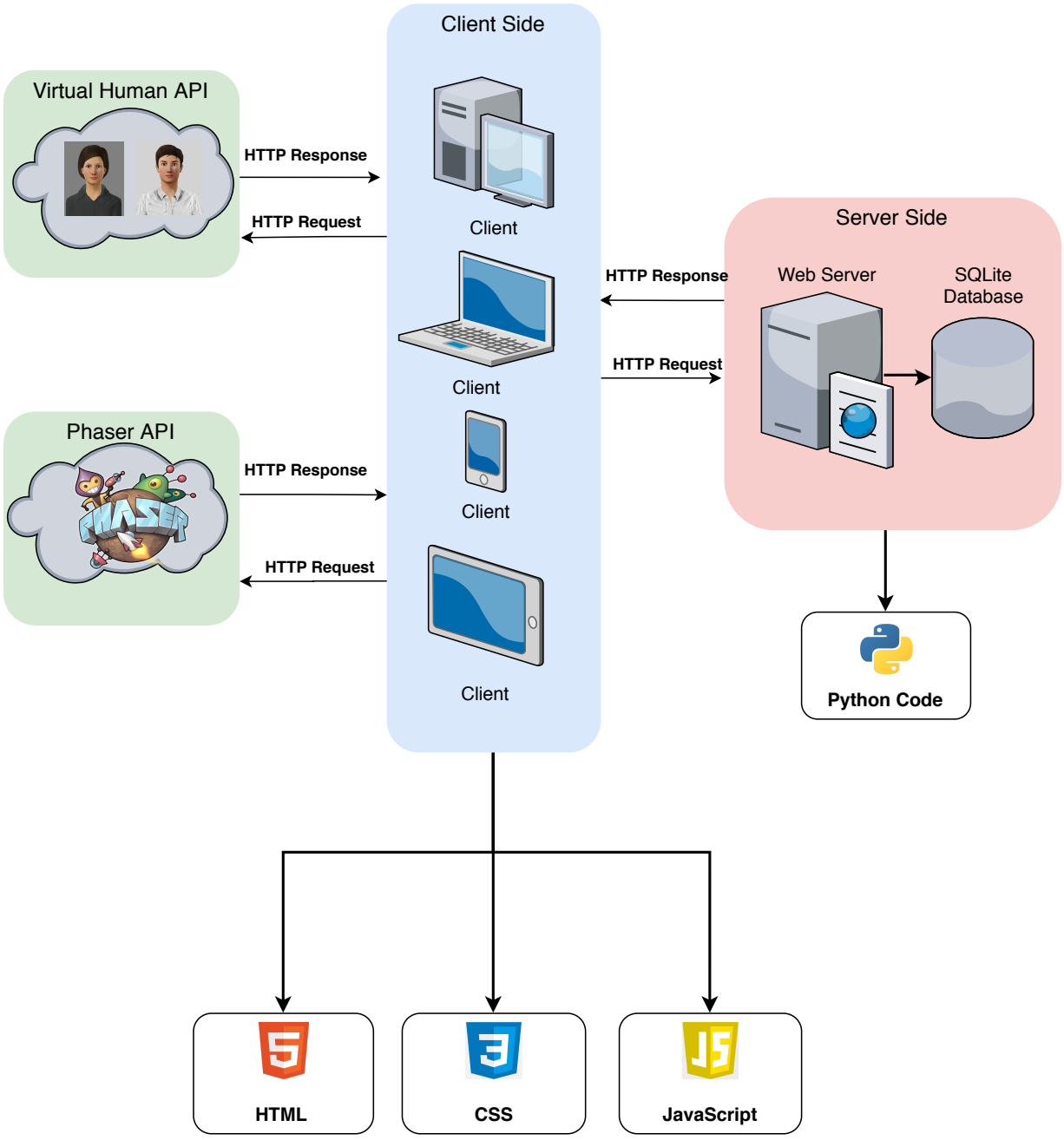


Figure 3.4: System overview of our prisoner's dilemma game architecture

3.2.1 Back-End

Web Server

The back-end consists of a web server that is able to process Hypertext Transfer Protocol (HTTP) requests and a SQLite database. The server is multi threaded Python web server built using the Python3. We added support for processing HTTP requests such as POST and GET. by adding HTTP method implementations to our request handler class.

Data Storage

Since we planned on running an experiment with human participants we needed some form of data or database storage to log and store in game data and as well as the data collected from the participant questionnaires. We decided to use a database engine known as SQLite. SQLite is a server-less SQL database engine and is unique in the way that it does not require running a separate server process [71]. More specifically, it is a embedded database meaning that it does not need to be running independently as a separate process [71]. As our server-side code implementation was written in Python we opted to use Peewee[1]. Peewee is an object relational mapping (ORM) tool that comes with support for SQLite making interaction with SQLite simple to use. Peewee made things easier for us as we were able to make updates and changes easily in Python. Most importantly, we can manipulate and treat the data collected as objects.

3.2.2 Virtual Human API

In order to study the effectiveness of emotionally inclined agents, we decided to integrate a virtual human into the gaming interface in which the human user would interact with while playing the game.

A virtual human would allow us to test our emotionally inclined artificial agent and in short enable us to put a face to our agent which would enhance the overall user and interactive experience. Studies have shown that humans prefer to interact with virtual humans in a similar way that they would interact with humans [32, 78]. It has been shown that virtual humans should be emotionally expressive and interactive in order to be considered effective for human interaction [78]. By utilizing a virtual human, our emotionally inclined

[1]http://docs.peewee-orm.com/en/latest/

(a) Female Virtual Human

(b) Male Virtual Human

Figure 3.5: Female and male virtual human

agent will be able to deliver both verbal and nonverbal cues throughout human interactions. Thorisson *et al.* has shown that non-verbal cues are of greater significance [97] . In addition to verbal cues such as speech, the non-verbal cues includes facial expressions, gestures, head movement, looking at the user and additional behaviours that support and enhance conversation.

The virtual human that we have integrated into our web application has been developed by a group from the Interactive Assistance Lab[2] located at the University of Colorado at Boulder [101]. The Interactive Assistance Lab works on designing and developing interactive learning solutions that uses photo-realistic virtual agents that can communicate via speech. [101]. It has also been used in applications relating to assistive technologies [59].

The virtual human toolkit comes with a simple JavaScript API making the integration into a web application possible. In order to instantiate a virtual human object onto the HTML web page, we pass a id referencing the tag in which the virtual human will then be injected into. An example of creating a virtual human instance is shown in Figure 3.6. The arguments to the create method which injects the virtual human into the div

[2]http://interactive.colorado.edu/

specified by the id are shown on line 9 of the code snippet in Figure 3.6. Figure 3.5 shows both female and male virtual humans. The resolution represents the quality of the model and currently supports 25%, 50%, 80% and 100%. Once the virtual human instance has been instantiated, we can then interact with it through the functionality provided through the virtual human API. Below we will go through some examples demonstrating basic usage, functionality, and capabilities of the Virtual Human Toolkit.

Virtual Human Usage

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Creating a Virtual Human Instance</title>
5    </head>
6    <body>
7      <div id="animate2"></div>
8      <script>
9      // cu_assistive.doCreate(div id,width,height,gender,
           resolution);
10     function CUAssistiveInit(){
11         cu_assistive.doCreate("animate2",470,480,"female",1);
12     }
13    </script>
14   </body>
15 </html>
```

Figure 3.6: Example of creating virtual human instance

Facial Expressions

The virtual human API comes with 6 provided methods that enable the usage of facial expressions. Figure 3.7 contains an example where the ‘cu_assistive.doSmileExpression() method is called to construct a Happy expression. The first parameter is the div id that references the div element containing the virtual human object and the second parameter percentage ranging from 0 to 100 specifying the strength of the expression. This strength value is scaled down to the range $[0, 1]$ when passed as an argument.

(a) Happy Expression at 20% strength

(b) Happy Expression at 90% Strength

Figure 3.8: Female virtual human happy expression

```
1    cu_assistive.doSmileExpression('animate2',0.9);
```

Figure 3.7: Construct happy facial expression at strength 90%

In Figure 3.8 two levels of the happy expression are displayed. Figure 3.8a shows a happy facial expression at 20% strength and Figure 3.8b shows a happy facial expression at 90% strength. Figure 3.9 shows 6 available API methods that are used to configure the virtual human's facial expression. These 6 methods correspond to the 6 universal emotions, happy, sad, disgust, fear, anger, and surprise. It is important to note that these facial expressions can be used in combination of each other to construct more complex facial expressions. This will be discussed in the following Chapter 4.

```
1    cu_assistive.doSmileExpression('div id',strength);
2    cu_assistive.doSadExpression('div id',strength);
3    cu_assistive.doDisgustExpression('div id',strength);
4    cu_assistive.doFearExpression('div id',strength);
5    cu_assistive.doAngerExpression('div id',strength);
6    cu_assistive.doSurpriseExpression('div id',strength);
```

Figure 3.9: List of methods to construct facial expressions representing the 6 universal emotions(happy, sad, fear, disgust, anger, surprise)

3.2.3 Game Development Framework

In this work, we designed and developed a web based application in the form of a web browser game. We came across a few different JavaScript game engines/frameworks that provided some key insights. However, in the end we decided to use Phaser.io[3] as it has a well documented API written in JavaScript and proved to be developer friendly[93]. It is also the most starred game frameworks on GitHub [93]. Phaser is an open source HTML5 game framework that provides both WebGL and canvas rendering supported through desktop and mobile web browsers [93].

Our aim is to design and develop a game that would be robust and would perform well for our research purposes. We outline the basic requirements of our system below.

For our experimental research, we planned on hosting our web gaming application online where participants would access the game remotely. In order to accomplish this we needed to learn and follow Phaser best practices and principles in order to make the overall experience seamless and easy to use.

In this section we will briefly present our design, structure, and some implementation details regarding our approach to building the Iterated Prisoner's Game with Phaser.

HTML5 Game Framework: Phaser

Phaser games are typically organized into Phaser states. Phaser states are objects containing a set of well named functions in which the game engine will know to call [21]. In other words you can refer to a state as a Phaser state object.

[3]https://phaser.io/

```
1
2     var phaserState = {
3
4         init:function(){
5         // Game setup to be executed before the game is loaded
6         },
7
8         preload:function(){
9         // Load in game assets that are needed for game. This
              includes images, audio files, etc.
10        },
11
12        create:function(){
13        // Setup up the game state with objects and assets that
              have loaded in.
14        },
15
16        update:function(){
17        // Game code logic and game loop. This code will be
              executed repetitively.
18        },
19
20        shutdown:function(){
21        // Any last minute code to be executed before Phaser
              state is over
22        }
23
24    }
```

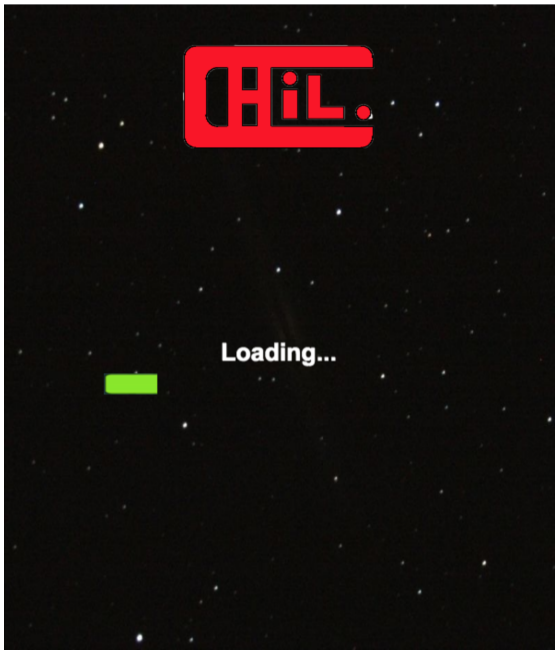
Figure 3.10: Example of basic Phaser state

One of the most important aspects to learning to use a game engine is to understand the game loop. This is essentially a loop that runs the entire game. This is necessary because in games there needs to be fluidity and uninterrupted motion and as a result some parts of code need to be executed around 20-30 times per second [21]. This is accomplished in Phaser through a update method that the game engine aims to call 60 times per second as shown in Figure 3.10 [93].

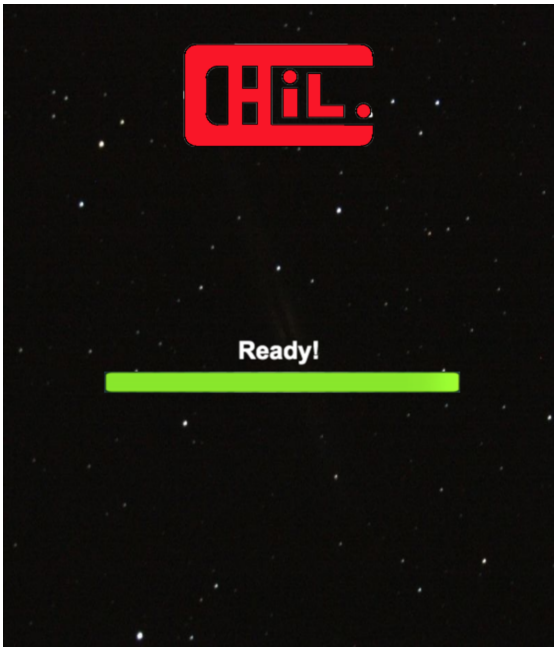
Our prisoner's dilemma game is organized into multiple states. The multiple states

are the boot state, splash state, menu state, and main state. These four states make up the core of the prisoner's dilemma game and are each written in individual JavaScript files. These 4 files or, also referred to as states contain a majority of the client side code for the game. Figure 3.11 contains the sequence of states that are executed in order to start the game. Figure 3.11a is the splash screen state in which all of the in game assets needed for the prisoner's dilemma are loaded into the game. The reason for the splash screen is primarily to display load progress to the user. This is a common practice in game development as loading assets may take some time until this process is completed [21]. Therefore, it is important to provide some sort of feedback to users regarding the status of the load time. We have utilized a graphic bar that is scaled from 0% (invisible) loaded to 100% as more assets are loaded. However, as the splash screen itself has assets such as the loading bar graphic and background, there is an additional state that is initiated before the splash screen commonly referred to as the boot state [21]. The boot state executed very quickly as the assets only necessary for the splash screen are loaded in at this time and the transition from the boot state to the splash screen is often unnoticed by the user.

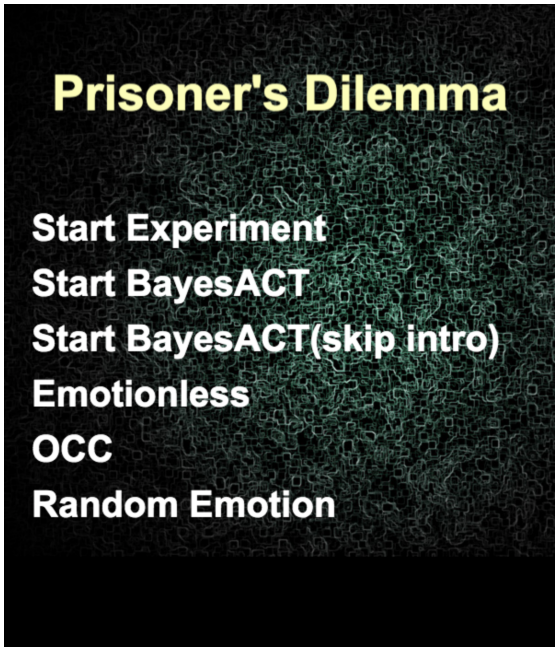
After the game assets are loaded the splash screen indicates to the user through a complete full loading bar and a message stating "Ready!" as shown in Figure 3.11b. Immediately following, the game menu state is then started as shown in Figure 3.11c. We have various options on the game menu which contain different configurations of our prisoner's dilemma game. These are essentially different "game modes". As we have different agents in the back-end, we have options pointing to the various agent implementations. However, for our experiment all of the menu options are hidden and there is a single option "Start Experiment" discussed later in Chapter 6. For development and testing purposes we needed a way of accessing different game configurations and we were able to accomplish this with having menu options pointing to each game configuration or "game mode". A menu option is selected by clicking one of the options on the game menu screen. Once selected the main game state will be initiated based on the configuration selected, as shown in Figure 3.11d.



(a) splash screen with game loading



(b) splash screen with game finished loading



(c) game menu

(d) game loaded after clicking one of the menu options

Figure 3.11: game start up sequence

Chapter 4

Affective Facial Expressions for Virtual Humans and Analysis

4.1 Introduction

The expression of emotion serves as an important social function in human interaction. Non-verbal cues conveyed through facial expression is described as a modality for human communication and is fundamental aspect in social interaction[53]. It has been estimated that humans can make and recognize as many as 250,000 facial expressions [9]. Studying the interpersonal effect of emotion within the context of social dilemmas has resulted in findings suggesting that emotions expressed through agents can influence human decision making[15]. Studies have shown that humans are sensitive to variations in emotion conveyed through facial expressions of an embodied agent. Melo *et al.* has studied the effect of emotion in the iterated prisoner's dilemma conveyed through facial displays of an embodied agent, and demonstrate there to be a resulting effect on human cooperation[16]. Therefore, it is imperative to design embodied agents that can effectively construct facial expressions representative of their emotional state.

In our prisoner's dilemma system we have two affective signals being transmitted from a virtual human. That is, facial expressions and utterances. As a result, we have designed and implemented a mapping from emotion to phrases (discussed in Chapter 5) and methods to map dimensional models of emotions to virtual human facial configurations.

In this chapter, we present three methods for facial expression construction in virtual humans. The first method is based off of conventional linear algebraic principles. The

latter two methods involves machine learning approaches, and deep learning architectures. Additionally, we provide further analysis into into what is being learned by these predictive models for valence and arousal dimensions.

4.2 Background

We present an overview into background materials such as a review of the current databases used in dimensional models of emotions, evaluation metrics, and facial expression parameters needed in the configuration of virtual human facial expressions.

4.2.1 Virtual Human Parameters for Facial Expression Configuration

In order to design a mapping for virtual human facial expressions, we identify the parameters needed to configure the virtual human. As discussed before, the basic functionalities of the virtual human API used in this work can be found in Chapter 3. In this section, we will provide a more detailed overview of what the configuration entails before discussing our algorithmic approaches.

Virtual human facial expressions are generated through three controls that map to specific sets of facial muscles into correlated and recognizable patterns. We refer to this three dimensional space of control as the happy-surprise-fear (HSF) space:

- Control 1: happy - sad
- Control 2: surprise - anger
- Control 3: fear - disgust

These specific sets of facial muscles are shown in Table 4.1.

Emotion	Action Units	Description
Happy	AU6 + AU12	Cheek raiser and lip corner puller.
Sad	AU1 + AU4 + AU6 + AU15	Inner Brow Raiser, Brow Lowerer, Cheek raiser and Lip Corner Depressor
Surprise	AU1 + AU2 + AU5 + AU15	Inner Brow Raiser, Outer Brow Raiser, Upper Lid Raiser and Lip Corner Depressor
Anger	AU2 + AU4 + AU6 + AU15	Outer Brow Raiser,Brow Lowerer, Cheek Raiser and Lip Corner Depressor
Fear	AU1 + AU2 + AU5 + AU15	Inner Brow Raiser, Outer Brow Raiser, Upper Lid Raiser and Lip Corner Depressor
Disgust	AU4 + AU6 + AU9 + AU10	Brow Lowerer, Cheek Raiser, Nose Wrinkler and Nose Wrinkler

Table 4.1: Facial action units mappings to the correlating facial muscles used in the virtual human API. The action unit (AU) is a high-level feature descriptor which describes the fundamental actions of individual or groups of facial muscles.

Although the virtual human face can be controlled by moving individual muscles, such as the inner eyebrow raise, groups of these are highly correlated and move in recognizable patterns. Therefore, these three dimensions of musculature movement are deemed sufficient. Therefore, a facial expression configuration for the virtual human involves the generation and setting in HSF space.

4.2.2 Facial Expression Databases

For affect recognition tasks, there are many facial expression datasets commonly utilized and publicly available. These datasets contain static facial images or facial image sequences and typically fall into two categories: posed and spontaneous. Many of these databases are used in training machine learning algorithms for applications involving automatic affective recognition. A majority of these databases are usually annotated based on categorical emotion models. JAFFE [56] is a database containing static images of posed facial expressions labelled categorically from six emotions. These six emotions happy, sad, angry, fear, disgust, and surprised form the construction of the six universal emotions which was proposed by Paul Ekman[20]. To name a few, the Cohn-Kanade [42], the extended Cohn-Kanade

dataset, [55], MMI [72], and Bosphorus [84] are all commonly used categorically labeled facial expression databases containing posed facial images and image sequences. BU-3DEF [103] is a database containing both posed and spontaneous facial images. FER2013 [30] is a database that is annotated categorically containing spontaneous static facial images.

Previous work regarding the release of annotated datasets containing facial images have often been conducted on much smaller scales and are often heavily focused on categorical emotion models. Categorical models of emotions are limited to representing a discrete subset of emotions. These databases often contain smaller samples and have been generated in controlled environments [18]. As a result, the use of dimensional emotion models are becoming more prominent as they capture more subtle and complex emotions. To name a few, some of the most commonly used databases containing image sequences in the form of video include RECOLA[79], SEMAINE[60], SALDB[19], and AFEW-VA[48]. The Aff-Wild [105, 45] and its extension Aff-Wild2 [46] are in-the-wild databases containing over 300 videos annotated on valence and arousal dimensions.

AffectNet

The AffectNet database [66] is used in our research as the primary source of data. Learning from dimensional models of emotion is ideal as more complex subtleties can be captured by trained models used in affect recognition tasks. AffectNet is currently the most complete and comprehensive database available for research in automated affective facial expression recognition[66]. AffectNet is the largest database containing facial images of both categorical and dimensional models of affect in the wild. The AffectNet database contains more than 1 million facial images compiled by querying three major search engines with over 1,250 emotion related keywords. These keywords were also queried in six different languages. Approximately half of the facial images have been annotated both categorically and as well as with valence and arousal dimensions. It should be noted that test set has not been released by the authors. As suggested by the authors, we have treated the validation set as our test set in our experiments. Our training set contains 320,739 images and the test set contains 4,500 images. In our experiments, we do not utilize the landmarks and instead train directly on the images.

The AffectNet database provides for each facial image in the wild, the following annotations:

- Location of 68 facial landmarks

- Seven emotion categorical labels (happiness, sadness, surprise, fear, disgust, anger, contempt)
- Valence and arousal values ranging from [-1,+1] of the facial expressions in the continuous domain

Below we present a few different visualizations to better understand and make sense of the distribution of the AffectNet database. A kernel density estimate which estimates the distribution of the training set is shown in Figure 4.1. Figure 4.2 contains a histogram using hexagon binning of the annotated images in the AffectNet database. The AffectNet database almost captures the entire valence-arousal dimensional space and is shown in Figure 4.2. As the distributions shown in Figure 4.2 densely form a circular shape over the 2-dimensional space, it is apparent that the database does not contain very strong samples. That is, samples where valence is 1 or -1 and with arousal being 1 or -1. Nonetheless, all databases have their own inherent advantages and drawbacks. For our purposes utilizing the AffectNet Database presents a reasonable starting point as it is one of the more prominent databases used in dimensional models of emotion.

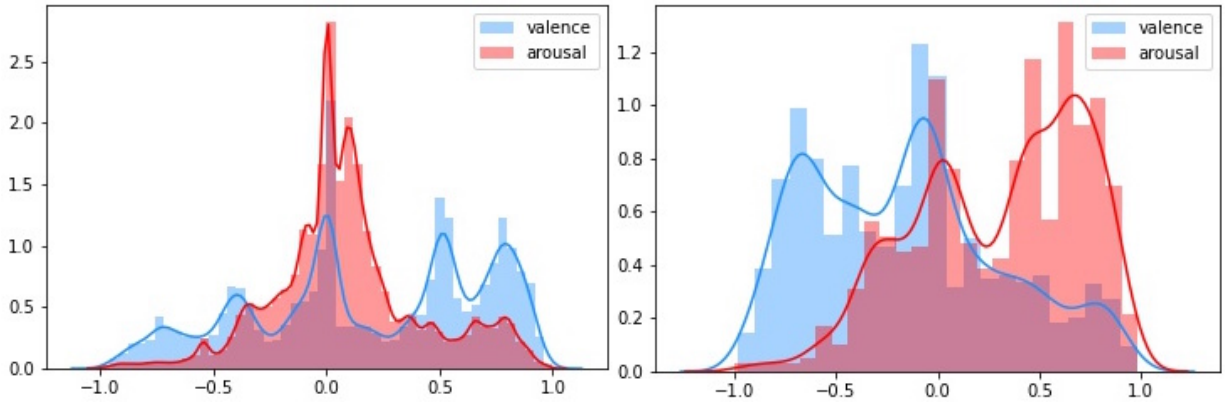


Figure 4.1: Training (left) and validation (right) set distributions of the AffectNet database with valence (blue) and arousal (Red) dimensions.

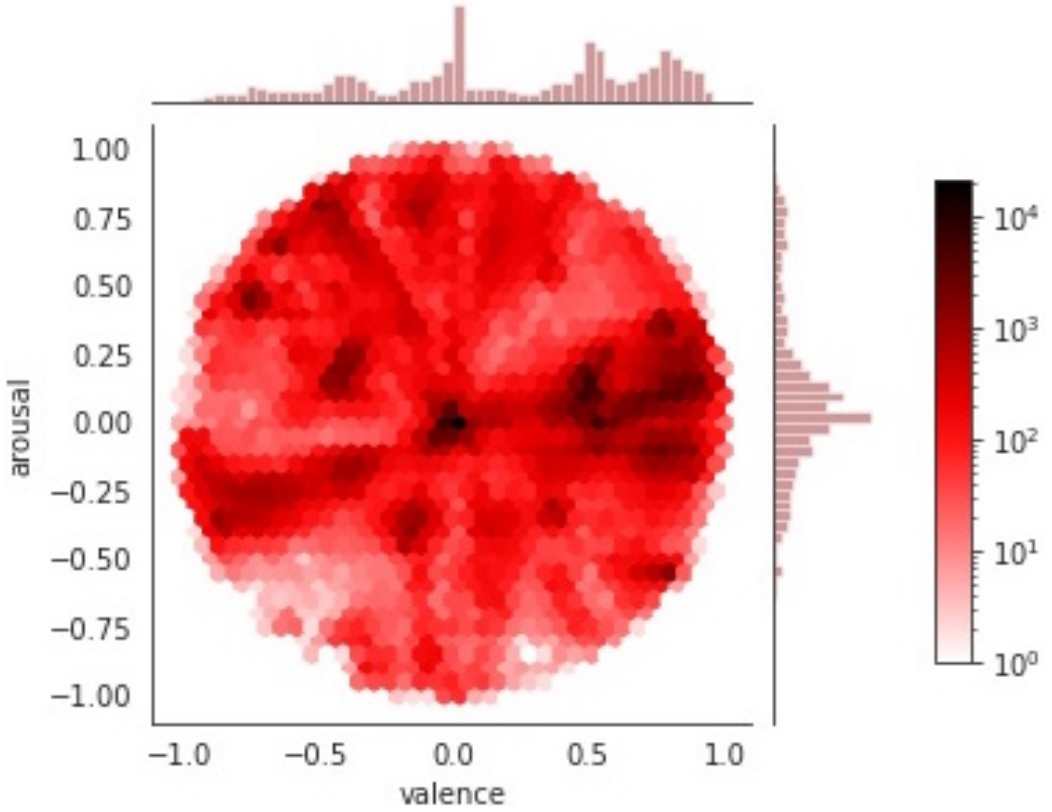


Figure 4.2: Distribution of AffectNet database. Histogram showing the annotated images of the AffectNet database. The density is captured and displayed by binning. Hexagon binning is used instead of square binning as hexagon binning has been shown to have more accurate data aggregation around the center of the bin and as a result we show both visualizations.

4.2.3 Evaluation Metrics

In part our work, we train machine learning algorithms to predict valence and arousal dimensions relating to affect. As this is a regression task, we have utilized metrics that are relevant and commonly used in dimensional models of the continuous domain. Below we summarize a few of the commonly used metrics used as part of our evaluation. All metrics presented below produce outputs within range [0,1].

Root mean squared error (RMSE) is defined by the following formula shown in Equation 4.1.

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(\hat{\theta}_i - \theta_i)^2} \tag{4.1}$$

From Equation 4.1, $\hat{\theta}_i$ is the predicted value of valence or arousal of the i^{th} sample, θ_i is the ground truth value of valence or arousal of i^{th} sample, and n is the number of samples in the evaluation or test set. As we are computing an error, the lower the RMSE, the better.

Pearson's correlation coefficient is defined in Equation 4.2. The Pearsons' correlation will measure how closely two related variables are in a linear manner. The higher the correlation value, the better it is for our task. From Equation 4.2, $\hat{\theta}$ are the predicted values of valence or arousal, and θ are the ground truth values of valence or arousal.

$$CC = \frac{COV(\hat{\theta}, \theta)}{\sigma_{\hat{\theta}}\sigma_{\theta}} \text{ where } \sigma_{\hat{\theta}} \text{ and } \sigma_{\theta} \text{ are standard deviation of } \hat{\theta} \text{ and } \theta, \text{ respectively.} \tag{4.2}$$

Equation 4.2 can also be written in terms of mean and expectation as shown below.

$$= \frac{E[(\hat{\theta} - \mu_{\hat{\theta}})(\theta - \mu_{\theta})]}{\sigma_{\hat{\theta}}\sigma_{\theta}} \tag{4.3}$$

The concordance correlation coefficient (CCC) is a metric that is also used and is shown in Equation 4.4 where $\sigma^2_{\hat{\theta}}$ and σ^2_{θ} represent the variances. $\mu_{\hat{\theta}}$ and μ_{θ} represents the mean values. CCC measures the agreement between the predicted values and ground truth. Consequently, it indicates whether the prediction matches the annotation. The higher the CCC the better it is.

$$CCC = \frac{2\rho\sigma_{\hat{\theta}}\sigma_{\theta}}{\sigma^2_{\hat{\theta}} + \sigma^2_{\theta} + (\mu_{\hat{\theta}} - \mu_{\theta})^2} \tag{4.4}$$

Given the nature of dimensional emotion prediction, a sign agreement metric (SAGR) is another metric that is effective in evaluating valence and arousal prediction models. This is shown in Equation 4.5. This metric is useful especially in tasks requiring emotion recognition. In emotion prediction tasks of valence or arousal, signs are essential. For instance, if the ground truth value is 0.2 for valence, a predicted valence value of 0.6 is

much better than a predicted valence value of -0.2. Despite the RMSE being the same in this example, 0.2 is a much better prediction as a positive valence prediction would still indicate a positive emotion which is much more aligned with the ground truth value. The higher the SAGR metric the better.

$$SAGR = \frac{1}{n}\sum_{i=1}^{n}\delta(sign(\hat{\theta}_i, sign(\theta_i)) \tag{4.5}$$

In Equation 4.5, δ is the Kronecker delta function defined as:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \tag{4.6}$$

4.3 Machine Learning Approaches

In this section, we will provide an overview and background relevant to our machine learning approaches pertaining to work presented in this thesis.

4.3.1 Histogram of Oriented Gradient

Histogram of oriented gradient (HOG) is a low-level feature descriptor that is commonly used in computer vision to extract features from images [13]. A feature descriptor can be described as an encoded representation of a data sample, in which important attributes have been captured. Other types of feature descriptors include Felzenszwalb HOG (FHOG)[23] which is a variant of HOG and as previously mentioned, the AU which has been codified into the facial action coding system (FACS) [24]. In our work, we train one of our baseline support vector regression model top predict valence and arousal using extracted HOG feature descriptors.

Figure 4.3: Example of HOG feature descriptor extraction shown on the right and the original facial image of a virtual human shown on the left.

HOG features are extracted from an image by first dividing the image into non-overlapping cell blocks. For all pixels within the cell block, gradients are computed, followed by the gradient magnitude and direction in both the x and y directions. After computing gradient vectors for each pixel, the gradient magnitudes are assigned to a 9-bin histogram based on the gradient direction. The histogram ranges from 0° to 180° and is divided into 20° partitions based on the orientation resulting in the 9 bins. A 9-dimensional feature vector is extracted from the histogram and used to represent the cell block. This process is computed for each block in the image. 36-dimensional feature vectors are formed by grouping 2×2 blocks together which are then normalized. Parameters such as the number of bins, cell block size, and block size can be determined empirically.

4.3.2 Support Vector Regression

The Support Vector Machine (SVM) is a supervised learning algorithm that is used in both classification and regression [35]. An SVM optimally constructs a hyperplane that separates the data, often in a higher dimensional space. In the application of an SVM, non-linear kernel functions are used to transform the data into a higher dimensional space making it linearly separable in the transformed space. This is achieved by the use of a kernel function which computes a similarity or relationship between pairs of data points.

The optimal separator will maximize the margin, meaning that the distance between each the closest instance of a class and the hyperplane is maximized.

For classification problems, the SVM set up is as follows [89]:

Suppose we have a set of data points $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$ where x_i is a feature vector and y_i is its corresponding label. A SVM can be constructed as a convex quadratic optimization problem [35] and can be expressed as optimizing the normal vector w to the hyperplane as shown in Equation 4.7.

$$\text{minimize } \frac{1}{2}||w||^2 \ , \quad \text{such that } \begin{cases} y_i - wx_i - b \leq \epsilon \\ wx_i + b - y_i \leq \epsilon \end{cases} \tag{4.7}$$

In Equation 4.7, ϵ represents a margin of error or tolerance. This indicates that errors are ignored as long they are less than ϵ.

Support Vector Regression (SVR) share similar principles with a few modifications [6, 89].

$$\text{minimize } \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \ , \quad \text{such that } \begin{cases} y_i - wx_i - b \leq \epsilon + \xi_i \\ wx_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{4.8}$$

In Equation 4.8, ξ_i and ξ_i^* represent slack variables [6]. In particular, this means that for any value that is outside of the range of ϵ, the deviation from ϵ is denoted by ξ. These deviations are added to the objective function shown in Equation 4.8. The constant parameter C controls the trade off between how many deviations of ϵ are tolerated and as well as the flatness (small w). This means that for larger values of C, the tolerance for points outside of ϵ increases. There are three common kernel types that are typically used. These kernel functions are linear, polynomial and radial basis function (RBF).

4.3.3 Principal Component Analysis

Principal Component Analysis (PCA) is a machine learning technique commonly used for dimensionality reduction [102]. PCA is a linear dimensionality reduction technique that aims to find principal components which best preserves the variance of the data. PCA, constructs orthogonal basis vectors which best captures the most important information

of the original data[1]. Principal components can be found with two methods where the first method involves finding the covariance matrix and the second method utilizes singular value decomposition (SVD) [96]. SVD provides a method to factorize a matrix into singular vectors and singular values. SVD construction is shown in Equation 4.9. $\boldsymbol{A}$ is decomposed as a product of three matrices, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, and $\boldsymbol{D}$ is a diagonal matrix containing the singular values of matrix $\boldsymbol{A}$.

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \tag{4.9}$$

We discuss the covariance method of finding PCA components. The covariance method of PCA is mathematically formalized as follows. Let $\boldsymbol{X}$ be the $N \times M$ input data matrix, where the columns $x_1, x_2, ..., x_N$ represent data samples and rows M, represent the features. Centering the data by the mean is calculated by subtracting the mean μ from each data sample in $\boldsymbol{X}$. The mean centered data matrix is defined below.

$$\boldsymbol{D} = \{d_1, d_2, ..., d_n\}, \text{ such that } d_i = x_i - \mu \tag{4.10}$$

The covariance matrix is calculated by $\boldsymbol{\Sigma} = \boldsymbol{D}\boldsymbol{D}^\top$, such that the variances are along the diagonal of $\boldsymbol{\Sigma}$. The covariance matrix, $\boldsymbol{\Sigma}$ is solved by calculating the eigenvectors and eigenvalues shown by Equation 4.11, where $\boldsymbol{V}$ and λ are the eigenvectors and eigenvalues, respectively.

$$\boldsymbol{V}\boldsymbol{\Sigma} = \lambda\boldsymbol{V} \tag{4.11}$$

Eigenvalues are scalar values and the non-zero eigenvectors are principal components such that each eigenvector represents a single principal component. Eigenvectors represent the directions of the PCA space and eigenvalues represent corresponding magnitudes. Therefore, the eigenvector with the highest eigenvalue represents the first principal component capturing maximal variance. The construction of the lower dimensional space is made by a linear combination of k selected principal components where maximal variance is preserved.

$$\boldsymbol{Y} = \boldsymbol{W}^\top\boldsymbol{D} \tag{4.12}$$

The lower dimensionality space is constructed by projecting the data onto the PCA space as shown in Equation 4.12, where $\boldsymbol{W} = \{v_1, v_2, ..., v_k\}$ for k selected components.

4.3.4 Artificial Neural Networks

Inspired by biological neural networks, an artificial neural network (ANN) is a machine learning model that provides a robust framework for supervised learning. One of the fundamental components of a biological neural network is the neuron, which receives, processes, and transmits signals to and from other connecting neurons. Similarly, an ANN contains a network of artificial neurons connected by weighted edges in which information flows through a series of intermediate computations. Neural networks aim to approximate some arbitrary function $f^*(\boldsymbol{x})$. While there are many variants of an ANN, the most common example is a feed-forward neural network, also known as a multilayer perceptron (MLP). An MLP is a directed acyclic graph, consisting of interconnected nodes in a layered structure.

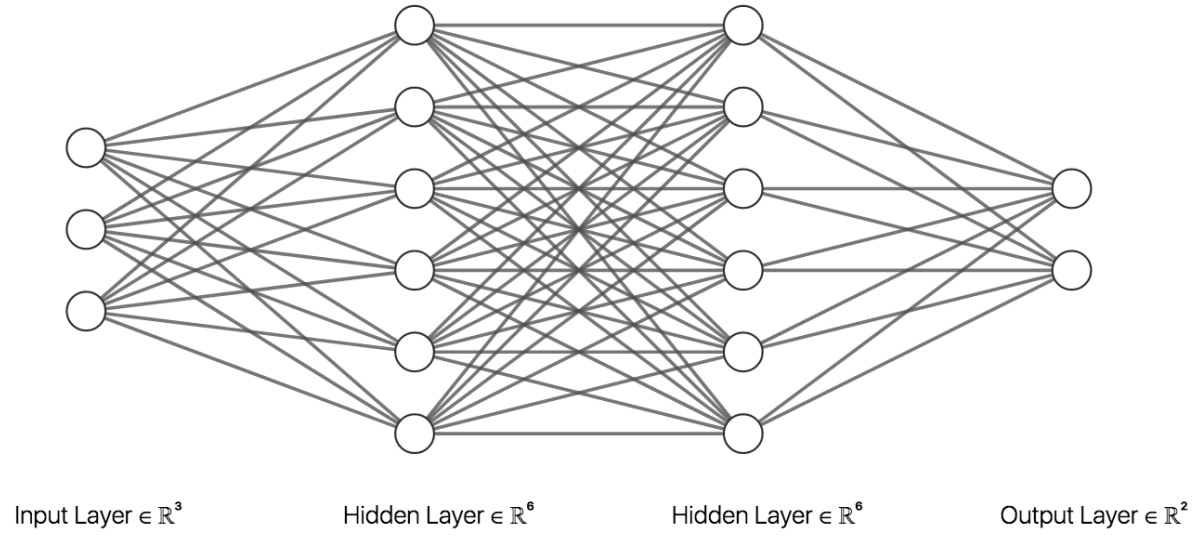


Figure 4.4: Feed-forward neural network with an input layer containing 3 nodes, 2 hidden layers containing 6 nodes in each, and an output layer containing 2 nodes.

An MLP learns the parameters $\boldsymbol{\theta}$ that best approximates the following function mapping $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$, where $\boldsymbol{x}$ is an input vector to the network parameterized by $\boldsymbol{\theta}$. Feed-forward neural networks are organized in groups of layers of interconnected nodes. The three layer types that form an MLP are an input layer, a number of hidden layers, and an output layer. Each node produces a real-valued output as the result of an activation function. Through a forward pass, the network processes one or more training examples

and adjusts the parameters such that the error is minimized. More precisely, this means given an input $\boldsymbol{x_i}$, the network outputs a prediction $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$. Given the ground truth labels $\boldsymbol{y_i}$, and the network's predictions $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$, a loss function $L(\boldsymbol{\theta})$ computes the errors. In the back-propagation algorithm partial derivatives of the loss function are computed with respect to the parameters. A learning algorithm known as gradient descent is then applied to adjust the parameters in a way which minimizes the error. The adjustment or update of the parameters is shown in Equation 4.13, where η represents the learning rate. The learning rate η, is a hyperparameter which controls the step size in the gradient descent learning algorithm. An iterative process of forward propagation and backpropagation continues until convergence is reached.

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{4.13}$$

Figure 4.4 is an example of a feed-forward neural network is shown with 4 layers with interconnected nodes.

4.3.5 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a specialized variant of deep neural networks which have been successful in many computer vision problems including image classification, object detection, and neural style transfer [28, 50]. As its name suggests, a CNN is a neural network which leverages the convolution operation. There three types of layers forming the building blocks of a CNN and are as follows.

1. Convolution layer
2. Pooling layer
3. Fully connected layer

Convolution Layer

The convolution layer consists of a **filter** also known as a **kernel**, in which the convolution operation is executed. In the convolution operation, we overlay the filter on top of the input image and compute an element-wise product and sum. The filter moves right and the element-wise product and sum is computation is repeated. The amount in which

the filter moves is specified by a hyperparameter known as the **stride**. The convolution operation is executed over the entire image, after which an activation function is applied. The final output of the convolutional layer is a feature map which may be passed to another layer in the CNN. The filter size is hyperparameter and typically of a much smaller size than the input image. For example, a 6×6 input image convolved with a 3×3 filter results in a 4×4 feature map assuming a stride of 1. Figure 4.5 shows an example of the convolution operation.

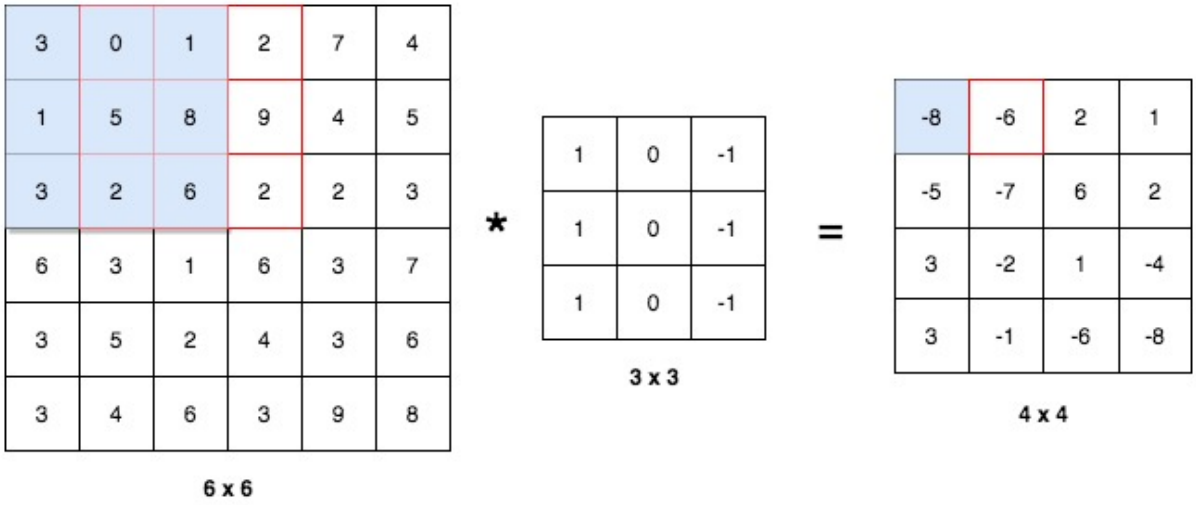


Figure 4.5: Convolution operation with a filter commonly used for vertical edge detection. The shaded blue region shows the convolution operation for the 3×3 region. In this example, -8 is the output resulting from the element-wise product and sum. The outlined red 3×3 region shows another example when the filter is shifted right with stride of 1. The resulting element-wise product and sum for the outlined red region is -6.

It is apparent, that the input image shrinks and the dimensions are reduced as it propagates further through the network. As a result, for deeper networks **padding** is used to slow the rate at which the input image shrinks. Another benefit of padding is that it enables the CNN to capture more information from the edge pixels as they are used more than once in the convolution operation. Common filters include the Sobel filter which results in an image that emphasizes edges, and the Scharr filter which finds vertical and horizontal edges [90]. However, in deep learning, these filters can be learned and treated as parameters instead of being hand crafted.

Pooling Layer

Pooling layers are used to reduce the size of the representation. Pooling can be considered as a way to compress, or more importantly preserve features that were highlighted by the convolution. Consequently, this will speed up computation.

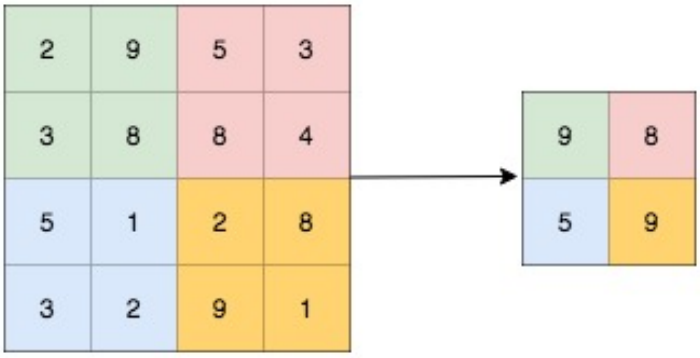


Figure 4.6: Max pooling with a stride of 2 and filter size of 2×2. The shaded regions represent the corresponding outputs of the max pooling operation.

Max pooling is a commonly used type of pooling in which an $m \times m$ filter is placed over a region of the input image and the maximal value of the block is extracted into the targeted downsampled matrix. Average pooling is another pooling method in which the average value of the block is extracted. Two hyperparameters associated with the pooling layer is the filter size and the stride.

Fully Connected Layer

In the layers preceding the output layer of a CNN, the feature maps are flattened into a one dimensional vector and propagated through one or more fully connected layers. The fully connected layers resemble exactly an MLP as shown in Figure 4.4. Fully connected layers are able to capture the relationships between high level features which are learned. The final output layer of the neural network is then constructed based on the problem, that is either regression, or classification.

4.4 Deep CNN Architectures

In this work, we modify and train two popular CNN architectures which have achieved state-of-the-art performance in many computer vision problems including object detection and face recognition to predict valence and arousal values. We begin by providing a brief overview of these deep CNN architectures.

4.4.1 Residual Networks

Deep networks consisting of many layers are difficult to train and thus presents several challenges. One example is the vanishing or exploding gradient problem. More precisely, this means when training deep networks, the gradients can get very small or very large. Another issue with deep networks is known as the degradation problem with respect to training accuracy. That is, as networks are designed with increased depth, accuracy has been shown to be saturated, followed by a degradation. Furthermore, it has been shown that degradation is not a result of overfitting, as adding more layers led to a higher training error [34]. Residual networks (ResNet) address these challenges by proposing a deep residual learning framework through a mechanism known as *skip connections* [34]. Skip connections take the activation from one layer and feed it to another layer deeper in the neural network. This means that features are being copied from earlier layers to much deeper layers. Skip connections do not add additional parameters or additional model computation complexity [34]. In short, a major benefit is that skip connections allow the backpropagation signal to reach from deeper layers to the input layers and have resulted in increased performance as learning this residual mapping is easier to optimize.

4.4.2 ResNet-50

In our training, we utilize the following ResNet-50 architecture shown in Table 4.2 [34]. He *et al.* initially trained ResNet-50 on the ImageNet dataset[17] with input image dimensions of 224×224×3. In this work, we also adopt the same input dimensions in our experiments. The input to ResNet-50 architecture is an image of dimensions 224×224×3. The output of the ResNet-50 architecture was originally used for classification of the ImageNet dataset containing 1000 classes.

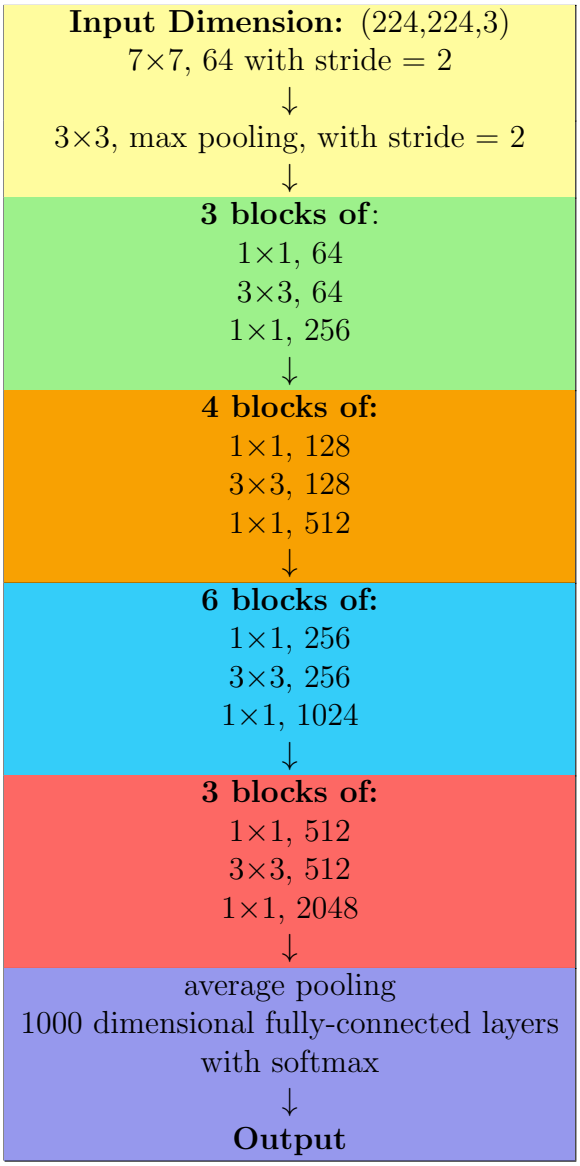


Table 4.2: The ResNet-50 architecture [34] consists of 4 main blocks of convolution layers. The first block consists of 3 repeating convolution blocks of 1×1, 64 → 3×3, 64 → 1×1, 256 with skip connections. The second block consists of 4 repeating convolution blocks of 1×1, 128 → 3×3, 128 → 1×1, 512 with skip connections. The third block consists of 6 repeating convolution blocks of 1×1, 256 → 3×3, 256 → 1×1, 1024 with skip connections. The fourth block consists of 3 repeating convolution blocks of 1×1, 512 → 3×3, 512 → 1×1, 2048 with skip connections. The input of the network is at the top (light yellow) block and the output of the network is the bottom (purple) block.The different colors indicate the different components making up the ResNet-50 architecture.

4.4.3 VGG-16

Extensive evaluation of networks of varying depths and utilization of small convolution filters (3×3) have been shown to have promising performance for depths of 16 to 19 layers. VGG-16 networks have been proposed and demonstrated that its representations generalize well to other datasets while achieving state-of-the-art results [88]. The idea is to stack convolutional and pooling layers in a systematic way in which the dimensions of the feature maps are decreasing while the channels are increasing. The underlying principle is that the filters double as more layers are added. We utilize the VGG-16 architecture shown in Table 4.3. The VGG-16 architecture also takes as input, images that are 224×224×3. In our experiments, we use the same input dimensions of when training on images.

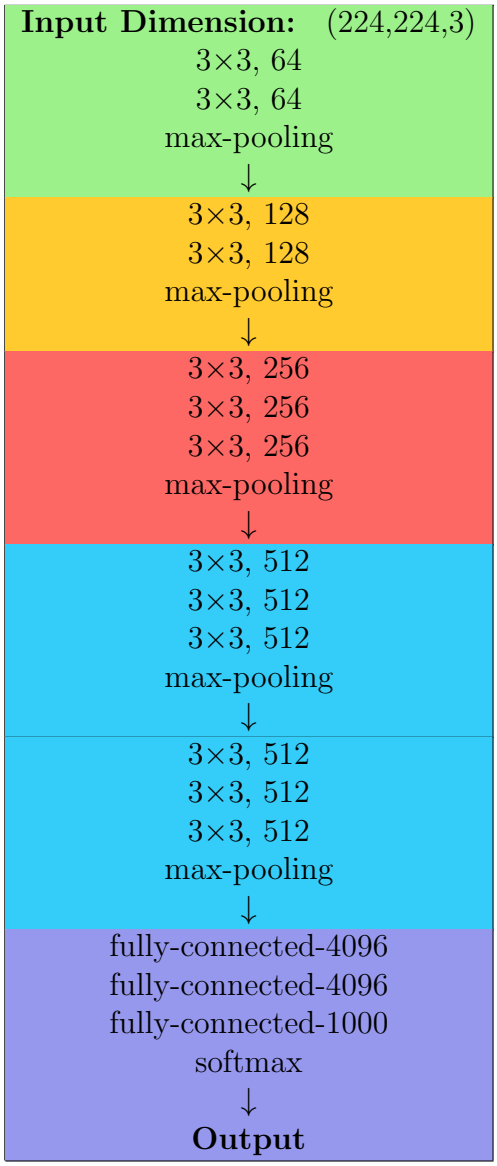


Table 4.3: VGG-16 CNN network architecture. The network schematics include 16 layers of learnable parameters including the convolution layers in which the number of filters double as the network increases in depth. Initially starting with 64 filters followed by 128, 256 and 512 filters reached at maximal depth. The last 3 layers are the fully-connected layers resembling a 3 layers MLP. Each colored box represents a fundamental component of the VGG-16 architecture. The input of the network is at the top (green block) and the output is the bottom (purple block).

4.4.4 Datasets for Deep Face Recognition

In this work, we adopt transfer learning by utilizing models that have been trained on two very popular face recognition datasets. We provide a brief description of these datasets below.

VGGFace

VGGFace [73] is a CNN with a VGG-16 architecture (discussed in Section 4.4.3) that has been trained on a facial dataset containing more than 2.6 million facial images. These facial images contain 2622 celebrities. This dataset has been used extensively in face identification and recognition applications.

VGGFace2

VGGFace2 [10] is a large-scale face dataset proposed as an extension of VGGFace. In this work a ResNet-50 (discussed in Section 4.4.2 and Section 4.4.1) was trained from scratch on the VGGFace2 dataset for face recognition. VGGFace2 contains 3.31 million images with 9131 subjects. On average there are 362.6 images for each subject. Images have been compiled by querying Google Image Search. One of the aims of the VGGFace2 dataset was to account for more variation with respect pose, age, ethnicity, and as well as to have more images for each identity.

4.4.5 Virtual Human Face Dataset

In this section, we describe a dataset in which we have constructed in order to learn the virtual human configurations in HSF space. Details regarding the virtual human API are discussed in Chapter 3. We generated 9,200 virtual human face images based on the HSF controls specified in Section 4.2.1. For all combinations, increments of 0.1 were utilized for each of the three dimensions. For each combination of controls, subsets of size 2 and size 3 were factored into the generation of the dataset. Our dataset contains 9,200 HSF configurations and their corresponding virtual human face image. To the best of our knowledge, our work is the first to generate a dataset of virtual face images and in turn use it to learn facial expressions for virtual human facial expression.

4.4.6 Transfer Learning

In our work, we apply transfer learning by crafting state-of-the-art models used in face recognition to dimensional affect recognition. We provide a brief description of this technique. Transfer learning refers to the improvement of learning a new task through the transfer of knowledge leveraged from an already learned task [98]. In recent years, the development of machine learning algorithms that facilitate the use of transfer learning has emerged as a prominent technique for addressing several challenges. To name a few, these challenges may include insufficient training data, lack of computational resources, or poor performance [94]. To address these challenges, deep learning researchers have utilized many techniques involving powerful state-of-the-art deep learning models trained on large-scale datasets. Transfer techniques include fine-tuning pretrained neural network models on a customized dataset, or using these models as feature extractors which would then be used in training a machine learning model.

4.5 Linear Algebraic Approach

In this section, we first present a facial expression configuration mapping based on algebraic principles. Recall that the problem is to map dimensional models of emotion to a virtual human facial expression configuration. That is, to map an EPA vector to the correct setting in HSF space.

4.5.1 Proposed Method I

Given an EPA vector, we can find the closest emotions corresponding to the HSF controls shown in Section 4.2.1 by using the ACT dictionary[1] as shown in Table 4.4. This is accomplished by first determining which three emotions in each dimension must be applied and then assigning a real number representing the intensity denoted $i \in [0, 1]$. More precisely, we compute the distance to each end point based on the vectors displayed in Table 4.4, take the closer of the two for each HSF control, and then normalize the distance to that point in order to compute i.

[1]EPA values taken from the (G)eorgia-UNC data 2015-2016

Emotion	Symbol	E,P,A
happy	h_+	3.4469, 2.9125, 0.2438
sad	h_-	-2.3793, -1.3414, -1.8759
surprise	s_+	1.4796, 1.3151, 2.3139
anger	s_-	-2.0267, 1.0667, 1.7967
fear	f_+	-2.4077, -0.7577, -0.6808
disgust	f_-	-2.5706, 0.2676, 0.4265

Table 4.4: Six universal emotions in the HSF space and their corresponding EPA values. EPA values taken from the (G)eorgia-UNC data 2015-2016.

4.5.2 Method I Details

Our algebraic approach is summarized in a two step process as follows.

Step 1: Find three Emotions to Apply
For some EPA vector $\mathbf{v}$, we use the Euclidean distance defined in Equation 4.14 to measure to determine if $\mathbf{v}$ is closer to h_+ or h_- as shown in Table 4.4[2]. The same distance measure is used for determining the remaining emotion dimensions (s_-, s_+, f_-, f_+). For example, if the EPA vector is closer to happy and further from sad, then for control 1 happy will be applied and sad will not be applied. Equation 4.14 computes the Euclidean distance, $d(p, q)$ between two n-dimensional vectors, $\mathbf{p}$ and $\mathbf{q}$. Figure 4.7 shows an example of how the happy dimension is determined to be applied given an EPA representing the emotion "hopeful". In Figure 4.7, happy is applied as the Euclidean distance to h_+ is lower than the distance to h. The other dimensions to be applied are determined in a similar manner.

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{4.14}$$

[2]EPA values taken from the (G)eorgia-UNC data 2015-2016 https://research.franklin.uga.edu/act/data-sets-0

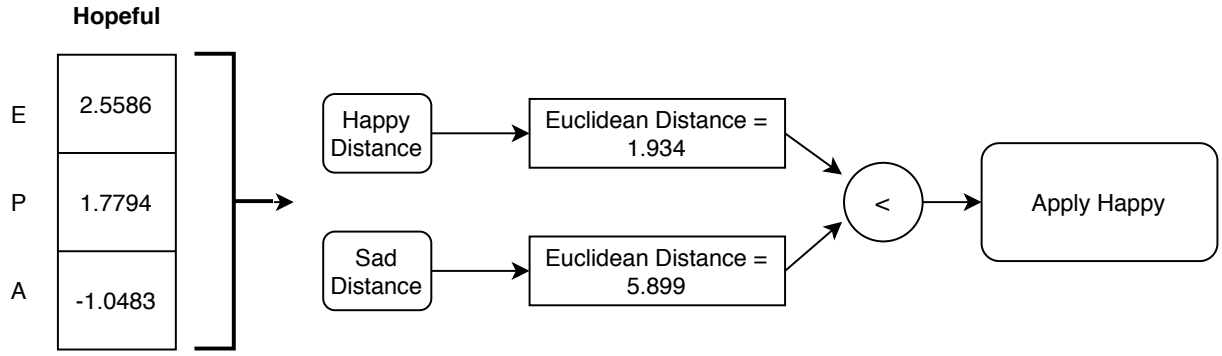


Figure 4.7: Example demonstrating that the happy dimension must be applied instead of sad. The query EPA represents "hopeful".

Step 2: Assign Numerical Value

The final step is to compute an intensity value which will be assigned to the three emotions. For example, suppose the EPA vector is closer to happy as opposed to sad. We assign a numerical value $i \in [0, 1]$ to happy. The closer the EPA vector is to happy, the stronger the displayed happy expression. The further away the EPA vector is from happy, the weaker the displayed happy expression. Based on the API specifications, this distance is mapped by a linear interpolation into the range [0,1] to specify a numerical value for the expression. Similarly, the same method is applied to the remaining dimensions (s_-, s_+, f_-, f_+). Note that when assigning a value in [0,1] to expressions, values closer to 1 will result in a stronger expression and values closer to 0 will result in a weaker expression. In addition, based on our experiments we achieved more naturalistic expressions when fear or disgust expressions are only applied when the evaluation (E) dimension of the EPA vector is negative.

4.6 Proposed Method II Formulation

In this section, we present our second method for constructing virtual human facial configurations in HSF space. This approach involves predicting valence and arousal dimensions from a facial image. After producing a trained predictive model, we predict valence and arousal of a virtual human facial image. Therefore, we construct a dataset of virtual human facial images and their corresponding configurations spanning HSF space. We generated our own dataset containing 9,200 virtual human facial images with their corresponding configurations in HSF space. This dataset is discussed in further detail in Section 4.4.5.

The steps involved in this approach are are as follows:

1. Train two machine learning models to predict valence and arousal dimensions, respectively. More precisely, we train a mapping from real facial images to predict valence and arousal. We experiment with a few machine learning algorithms. We evaluate the performance of each based on our metrics defined in Section 4.2.3 and utilize the model with the best overall performance. Based on our experimental results, we utilized our VGG-16b (described in Section 4.6 model for predicting both valence and arousal. The results for these models and other models in which we experimented with are displayed in Table 4.5 and Table 4.6.

2. Generate a virtual human facial image dataset containing corresponding configurations in HSF space. Apply the mapping from step 1 (using the superior model), to the generated set of virtual human facial images. That is, we now have a mapping from virtual human faces to valence and arousal.

3. Invert the mapping from step 2, to obtain a mapping of valence and arousal to the corresponding virtual human face configuration in HSF space. Valence and arousal are congruent to evaluation and activity, we linearly map these values into a range of [-4.3,4.3] consistent with EPA space.

Therefore, by generating and defining a set of virtual human facial expression configurations in HSF space, we have in turn devised a systematic approach to mapping emotion to a virtual human facial configuration. That is, given an evaluation and activity, we can efficiently query the closest virtual human face and apply the queried configuration in HSF space. In this method, the potency dimension is omitted and we only use the evaluation and activity dimensions. More precisely, we are mapping EA to HSF space. In this method, we make the assumption that applying the mapping trained in step 1 to virtual human faces is sufficient. This is because the AffectNet database is sufficiently diverse and spans nearly the entire valence and arousal space.

4.6.1 Method II Details

In this section, we present details regarding the models in step 1) described in Section 4.6.

Machine Learning Models for Dimensional Affect Recognition

We experimented with three machine learning techniques for predicting valence and arousal. We trained VGGFace (VGG-16 architecture), ResNet-50, and SVR models. In particular,

two separate models are trained for valence and arousal dimensions. Each model is trained on the AffectNet database described in Section 4.2.2. The input to our models were images from the AffectNet database and the output is a 2-dimensional vector containing the valence and arousal predictions. Our intention with training an SVR model was to use it as baseline for comparison with our deep CNN models as we expect deep CNNs to have superior performance.

In order to train our SVR model the following steps were executed. Images were resized to 256×256 pixels and HOG features were extracted with a cell size of 8. We then applied dimensionality reduction in an effort to ease computational complexity and optimize the time for training. This was accomplished by applying PCA and selecting the first k eigenvectors retaining 90% of the variance of the features. Two separate SVR models were trained to predict the valence and arousal dimensions. In our experiments, we use the *rbf* kernel and the implementation of SVR (with default parameters) in sklearn[3].

To train the deep CNN models we modified the VGG-16 and ResNet-50 architectures output layers to accommodate regression. That is, the final output layer consisted of a single neuron with a linear activation function. The output is a real value between [-1,1]. All CNN models were trained using Adam as the optimizer. Adam is a stochastic gradient descent algorithm which utilizes momentum by calculating a running average of first and second moments of the gradient [44]. The parameters for Adam used in our experiments as follows: learning rate $\alpha = 0.001$, $\beta_1 = 0.1$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We adopted transfer learning described in Section 4.4.6 which involved leveraging pretrained models such as VGGFace and ResNet-50 (pretrained on VGGFace2). Particularly, we utilize the existing VGG-16 CNN architecture used in VGGFace and freeze the layers inherent to the original model. We then remove the top layer and add two fully-connected layers. The first connected layer containing 512 neurons followed by the output layer containing a single unit. A similar modification was used in the ResNet-50 model. We denote this variant of the VGGFace architecture as VGG-16a as shown in Table 4.5 and Table 4.6. It is our intention that learning from face detection and recognition transferable to affect recognition in dimensional models of emotion. This is based off of the premise that models applied in both domains have to learn similar low-level and high-level features. Both of these models were trained for 20 epochs with a mini-batch size of 256 until convergence. In addition, we trained a second VGG-16 denoted VGG-16b in Table 4.5 and Table4.6, with initialized weights from VGGFace, but instead made all the layers were trainable (layers were not frozen). The only difference in the second VGG-16 model is that all layers were made trainable (unfrozen). All models were implemented and trained using Tensorflow 2.0[4]

[3]https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html
[4]https://www.tensorflow.org/

through a multi-synchronous approach. That is, a single-host, multi-device synchronous training involving the usage of 8 Tesla K80 graphics processing units (GPU). Our results are summarized in Section 4.8.

4.7 Proposed Method III Formulation

In this section, we present a formulation of our final and novel third method for learning affective facial expression configurations for virtual human displays. Our goal is to devise a robust approach for mapping any given EPA to a facial expression configuration in HSF space. More precisely, we are mapping evaluation and potency to HSF space as we are omitting potency. That is, we map EAs to facial expression configurations in HSF space. Intuitively, we base our approach off of the premise that in order to effectively generate affective facial expression mappings for a virtual human display, we first must learn and capture the features that are inherent in real human facial expressions. That is, the types of features that are intrinsic to human facial expressions in an affective context pertaining to emotion. Our approach is summarized in the following steps.

1. Generate face embeddings for both the AffectNet database face images specified in Section 4.2.2 and the virtual human face dataset discussed in Section 4.4.5. We define a face embedding as a compact vector representation of a facial image. In our work, a *face embedding* is a vector of length 4096. We describe details of how we accomplish this in Section 4.7.1.

2. For each real face image in the AffectNet database, find the most similar virtual human face using the face embeddings generated from Step 1. The similarity metric is defined by the cosine similarity in Equation 4.15. The virtual human face which is most similar will be deemed most similar. We now have pairings of real face images and the most similar virtual human face.

$$\text{Cosine Similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.15}$$

   In Equation 4.15, $\mathbf{A}$ and $\mathbf{B}$ are each vectors representing face embeddings.

3. Based on the pairings from Step 2, construct a customized dataset by replacing the real faces back to their annotated dimensions of valence and arousal, and the virtual face to its configuration in HSF space. That is, (valence, arousal) $\rightarrow$ to HSF controls.

4. Train a feed-forward neural network model to learn a mapping from EA ratings to HSF controls on the customized dataset defined in Step 3.

In this approach, we are able to generate a virtual human facial expression in HSF space, given any evaluation and activity as input. We should note that since the range of valence and arousal is [-1,1], in order to predict the appropriate HSF controls we linearly map evaluation and activity values which range from [-4.3,4.3] to a range of [-1,1].

4.7.1 Method III Model Details

In this method, based on step 1 in Section 4.7, we compute a face embedding for all face images of the AffectNet database and all face images of the virtual human face dataset by modifying the VGGFace architecture in which only the first fully connected layer containing 4096 units is preserved. In the original VGG16 architecture displayed in Table 4.3, only the first fully connected layer is kept and the rest of the purple block is removed. Figure 4.8 shows the modified architecture of how we generate face embeddings for facial images. Intuitively, we leverage the power of discriminative features learned by VGGFace and are mapping a facial image to a more compact feature embedding through the modification of the VGGFace architecture.

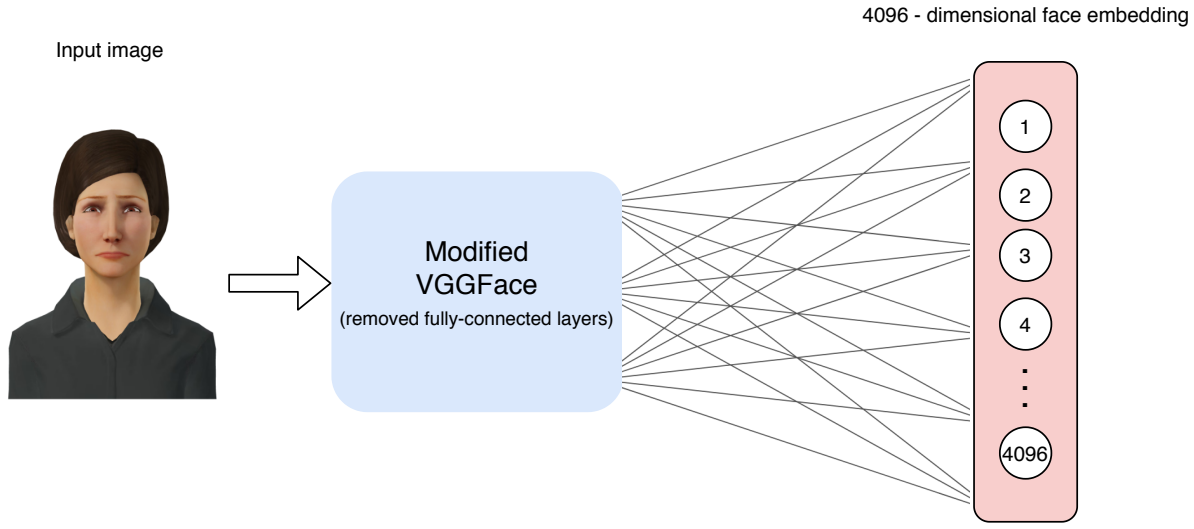


Figure 4.8: Example demonstrating a facial image being passed into our modified VGGFace model, with a 4096-dimensional feature vector as output. Only one fully-connected layer with 4096 units is preserved with the rest of the layers shown in the purple block in Figure 4.3.

From step 4 in Section 4.7, we utilize a simple feed-forward neural network as the regression model. We train a feed-forward neural network on the customized dataset defined in Section 4.7. The architecture of our neural network is as shown in Figure 4.9. We train our model for multi-output regression as the output we are predicting is the HSF configuration. Using the constructed dataset we defined in Step 2 of Section 4.7, we train this neural network to predict the HSF configuration. The model was trained for 20 epochs until convergence. We used the Adam optimizer with parameters: learning rate $\alpha = 0.001$, $\beta_1 = 0.1$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Using sklearn's model selection library, we split the training and validation data using a 0.8/0.2 split for training and validation sets respectively.

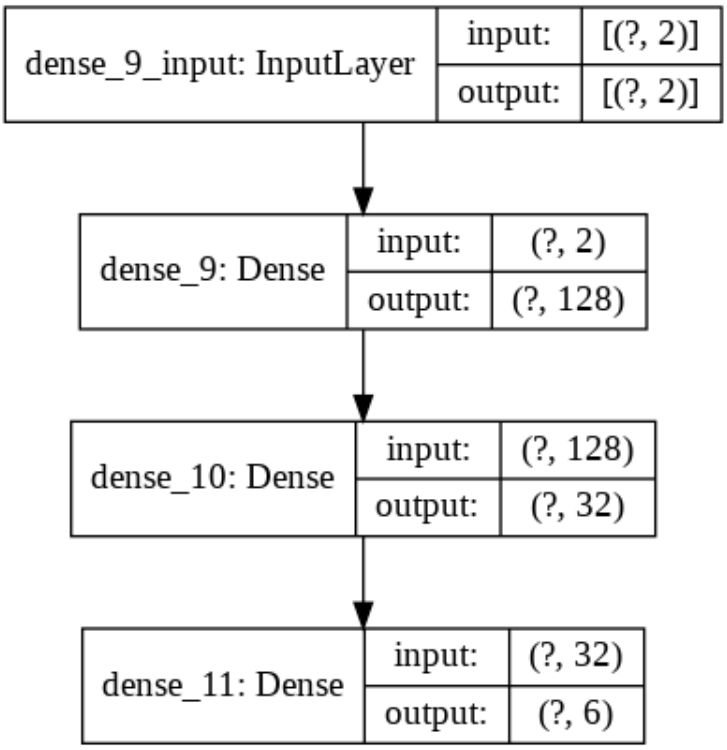


Figure 4.9: Feed-forward neural network with three layers. Input layer of two neurons for valence and arousal. A hidden layer containing 128 hidden units and an output later containing 6 units. The 6 units corresponding to the HSF space configuration.

4.8 Results

We present our results for methods described in this chapter. Table 4.5 and Table 4.6 show the results of the machine learning models we experimented with for method II. More specifically, in selecting

All of the models we utilize in this work for predicting valence are displayed in Table 4.5. As expected all deep CNN models outperformed the SVR models in valence and arousal prediction. This can be explained through the high variation in the training data enabling CNN models to learn more discriminative features as opposed to HOG features. As a result, the CNN models captured a better representation of the the valence and arousal dimensions. We observed that VGG-16b which was the model initialized with weights from a pretrained VGGFace model had outperformed the other models. We suspect that this is

because our training data size is large and that since all layers were trainable this led to more effective fine tuning for predicting valence. Our results for models trained to predict arousal are displayed in Table 4.6. Results for our models trained for predicting valence, indicate a similar improvement in the VGG-16b model for predicting arousal. We believe this improvement is explainable through similar reasoning with our predictive models for valence. For method II, we utilize VGG-16b models for predicting the valence and arousal for the virtual human dataset. Table 4.7 shows the validation losses for the feed-forward NN. We also report the results presented by the authors of AffectNet as follows[66]. For valence, RMSE = 0.394, CORR = 0.602, SAGR = 0.728 and CCC = 0.541. For arousal, RMSE = 0.402, CORR = 0.539, SAGR = 0.670 and CCC = 0.450. We note that these are the results on the unreleased test set.

Model	Valence **RMSE** ↓	Valence **MSE** ↓	Valence **CCC** ↑	Valence **CORR** ↑	Valence **SAGR** ↑
SVR	0.5087	0.2588	0.3132	0.3958	0.5864
ResNet-50	0.4491	0.2017	0.4402	0.5177	0.6932
VGGFace (VGG-16a)	0.4584	0.2106	0.4938	0.5225	0.6776
VGGFace (VGG-16b)	0.4324	0.1874	0.5141	0.5451	0.7173

Table 4.5: Results for trained valence models on test set containing 4,500 images from AffectNet. VGG-16a refers to the model where only the top layers was trainable and rest of the inner layers were not trainable(frozen). VGG-16b refers to the model where weights were initialized from pretrained VGGFace model and all layers were trainable. The range of the predicted valence values are [-1.1].

Model	Arousal **RMSE** ↓	Arousal **MSE** ↓	Arousal **CCC** ↑	Arousal **CORR** ↑	Arousal **SAGR** ↑
SVR	0.4488	0.2015	0.1456	0.2804	0.6715
ResNet-50	0.4119	0.1696	0.3124	0.4674	0.7034
VGGFace (VGG-16a)	0.4083	0.1670	0.3147	0.4594	0.7090
VGGFace (VGG-16b)	0.3988	0.1593	0.3950	0.4995	0.6930

Table 4.6: Results for trained arousal models on test set containing 4,500 images from AffectNet. VGG-16a refers to the model where only the top layers was trainable and rest of the inner layers were not trainable(frozen). VGG-16b refers to the model where weights were initialized from a pretrained VGGFace model and all layers were trainable. The range of the predicted arousal values are [-1.1].

Model	RMSE ↓	MSE ↓
Feed-forward NN	0.4018	0.1617

Table 4.7: Results for model III on test set using regression metrics defined in Section 4.2.3. We utilized a 0.8/0.2 split for training and testing sets as mentioned in Section 4.7.1.

4.8.1 Method Comparison

We compare each method by mapping a sample of EPA vectors to HSF space and generate the corresponding virtual human configuration. We now present some of our findings.

We select a small sample of emotion vectors and generate the corresponding virtual facial display based on the three methods presented in this chapter. Our results suggest that for emotions that have a negative evaluation, our method three is able to capture finer distinctions and subtleties. For instance, Figure 4.16 shows the virtual human face construction for resentful. We find that method III constructs a finer adjustment, improved and more accurate facial expression than method I. A similar finding is observed with other emotions with a negative evaluation. Figure 4.17 shows another example with the emotion"distressed". We also observe that method II has poor performance. We attribute this poor performance based on differences in their distributions. Precisely, the difference between a real facial image versus a virtual human facial image constructed with computer graphics. Additional comparisons are provided in Appendix A.

Figure 4.10: Method I

Figure 4.11: Method II

Figure 4.12: Method III

Figure 4.13: Resentful which is represented by the EPA vector [-2.02, -0.39, -0.9767]

Figure 4.14: Method I

Figure 4.15: Method II

Figure 4.16: Method III

Figure 4.17: Distressed which is represented by the EPA vector [-2.3886, -0.9171, 0.74]

4.9 Limitations

While we observe improvements in methods II and methods III, we must note that we are omitting the potency dimension in EPA space. As the AffectNet database is only annotated on valence and arousal dimensions, our methods were only able to account for the affective content captured by only the valence/evaluation and arousal/activity dimensions. In the initial construction of EPA space, potency has experimentally been shown to be a factor which explains the second most of the variance where evaluation is explains the most and activity explains the least [69, 70]. Therefore, our methods are limited in the sense that when we are mapping EA to HSF space, there is a loss of important affective information of the potency dimension. In method III, we assume that a face embedding is a sufficient representation of both real facial images and virtual facial images. We assume that measuring the cosine similarity between face embeddings of virtual humans and real facial images are sufficient. This is because, our modification to the VGG-16 model which generates a face embedding has been pretrained on the VGGFace dataset containing 2.6 million facial images. Based on the diversity and volume of the dataset, the VGGFace model can sufficiently capture relevant facial features and represent an image compactly as an embedding. We also propose a experimental evaluation procedure to measure our methods. There are not an clear metrics to evaluate our methods. Our methods can be evaluated by conducting a survey where participants can annotate the generated virtual human faces in EPA space. We would then be able to determine and establish consistency between the annotations and the EPAs used to construct the virtual human facial expression.

Chapter 5

Affective Utterances for AI Agent in the Prisoner's Dilemma Game

5.1 Introduction

Sentiment analysis is a study that analyzes people's opinions, sentiments, and emotions towards various entities [54, 27]. These entities could be services, organizations, issues and events [54]. Extracting affect from text and making sense of this information has garnered much interest in natural language processing. Sentiment analysis provides key insights into semantics and sentiments. One of the primary drawbacks of sentiment analysis is that most methods use a one dimensional approach, meaning that these traditional approaches are restrictive in terms of their expressiveness. Sentiment scores that are predicted are often assigned one of three labels. That is, positive, negative, or neutral. In essence, it does not capture or identify more complex human sentiments and emotion. These limitations present challenges when building AI systems. AI systems that can effectively engage in human interaction requires affective alignment with the ability to capture and interpret complex emotions. In this work, we leverage the EPA model (discussed in Section 2.4) and word embeddings, in an effort to develop more affectively aligned utterances for agents in the prisoner's dilemma game. We focus on the usage of word embeddings and devise a straightforward method of mapping emotions to text at the sentence level.

5.2 Background

This section presents general background information related to our method. We provide a brief overview natural language processing and word representations followed by a technique used to visualize high-dimensional data in a lower dimensional-space.

5.3 Natural Language Processing

Language is an integral part of human communication and interaction. In order to design and build AI agents that can effectively interact with humans, AI agents should be able to exhibit human-like communicative behaviour. As a result, machines that can understand language is key. Natural Language Processing (NLP) is a branch of AI in which computers are used to understand and analyze human language. For several decades NLP techniques have predominantly focused on more traditional methods that include manually designed rule based systems. In particular, traditional approaches in NLP focused on shallow methods involving machine learning techniques such as SVM and logistic regression [104]. Recent advances in machine learning, specifically deep learning, has resulted in tremendous progress in the field. For instance, deep learning methods based on vector representations have obtained superior performance in various NLP tasks including word embeddings. Deep learning approaches has outperformed many traditional approaches in NLP tasks such as Part-of-Speech tagging (POS), word lemmatization, Name Entity Recognition (NER), machine translation, and text generation [12, 104].

5.4 Bag of Words

Bag-of-words is a representation of text, where features represent single words contained in the training corpus. In this model, sentences are represented as a multiset of its words. Common preprocessing steps are applied in the application of the bag-of-words model. A common technique is to first filter out infrequent and frequent words. In addition, stop words, words which provide no additional informative content, are removed. These includes words such as pronouns, prepositions, and conjunctions [92]. Following the removal of these words from the vocabulary, a *stemming* algorithm may be applied to group morphological variants of the roots of words. Therefore, words with a similar stem will map to a common feature. For instance, after applying a stemming algorithm, the words $\{walker, walked, walking\}$, will map to the root word, *walk*. However, the bag-of-words

model presents a few shortcomings. One of the limitations of the bag-of-words model is that word combinations and order are often ignored, resulting in poor representations failing to capture the semantics of text [43]. Often times the vocabulary needs to be carefully designed, thus making the bag-of-words model a very time consuming and computationally expensive task.

5.5 Distributed Word Representations

Traditional NLP models are heavily based on hand crafted features. This includes n-grams, or one hot encoding representations. These features are expensive, time consuming and prove to be tedious to generate. In an effort to learn more meaning representations that can capture semantics, neural networks models have been proposed and proven to successfully learn distributed representations of words [7]. These neural networks are trained end-to-end and are able to learn a fixed dimensional real-valued vector space of words. Statistical language modelling approaches require the model to learn the distributed representation. That is, the similarity between words, and the probability function for sequences of words. Intuitively, this means that words that are used in similar ways will result in having similar representations in the learned dimensional space. Words with similar meanings share semantic and syntactic relationships captured in the neural network model.

5.5.1 Word2Vec

Word vectors are a mapped representation of words in a continuous vector space. As discussed in the previous section, words that share semantic similarities also share a similar vector representation. Word2Vec is one of the most widely used pretrained word embedding algorithm. Word2Vec uses a very shallow neural network to predict the context of words in which the context of words is optimized [65]. Distributed word representations for Word2vec can be learned through two approaches.

1. The **Continuous Bag of Words** (CBOW) model in which the model aims to predict the current word given the a window of surrounding context words.
2. The **Skip Gram** model in which the model is given the current word and predicts the surrounding window of context words.

Word2Vec embeddings have resulted in very powerful observations in the way that vector arithmetic can be applied. One of the most famous examples is the analogy of

"man is to woman as king is to queen" where the computation in Equation 5.1 solves for "queen".

$$king - man + women = queen \tag{5.1}$$

While syntactic, semantics, and co-occurrence statistics are captured from distributed word representations, these approaches present limitations. These approaches are not able learn and capture features pertaining to sentiments and emotion. This can be explained because words that share similar contexts, may not share similar sentiments. For example, "good" and "bad" may share similar contexts but differ in context. For instance, "The bad dog." and "the good dog." share similar contexts, but differ in their sentimental meaning. In our work, we propose a method that leverages the EPA space in order to overcome this drawback for generating utterances for a prisoner's dilemma agent.

5.5.2 Google Pretrained Word2Vec Model

Phrase generation for our prisoner's dilemma agent, relied on the use of word embeddings at the sentence level. Our approach involved mapping an emotion label to a word embedding. We utilized Google's pretrained Word2Vec model for all word representations. These word vectors contain 300 features and have been trained on Google news articles with a vocabulary size of 3 million.

5.6 t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique used for visualizing high-dimensional data [57]. This is accomplished by reducing the dimensionality in which each datapoint is assigned a location in two or three dimensional space. t-SNE is an improvement from the previous Stochastic Neighbor Embedding (SNE) as the visualizations are a lot more effective and proves to be easier to optimize [38]. The problem of dimensionality reduction involves the conversion of a high-dimensional data $X = \{x_1, x_2, x_3, ..., x_n\}$ such that $x_i \in R^d$ is mapped into either two or three dimensional space of the form $\boldsymbol{\gamma} = \{y_1, y_2, y_3, ..., y_n\}$ where $y_i \in R^2$ or R^3. The goal is to preserve as much of the important information in the reduced representation. This reduced representation can easily be used for visualization in two or three dimensional plots.

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{5.2}$$

The idea of t-SNE is to construct a probability distribution by the conversion of high-dimensional Euclidean distance between the datapoints into similarities represented as conditional probabilities. The similarity of datapoint x_j to x_i is denoted by the conditional probability $p_{j|i}$. This probability of similar data points can be modelled by the equation shown in Equation 5.2. For nearby points, $p_{j|i}$ is relatively high and for separated data points closer to zero . In Equation 5.2, σ_i is the variance of the Gaussian centered on the datapoint x_i. This probability distribution uses the Gaussian distribution only to capture relationships between points in the high-dimensional space.

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_i - y_l\|^2)^{-1}} \tag{5.3}$$

The two main modifications from the original SNE are as follows. t-SNE optimizes a "symmetrized" SNE cost function and the recreation in low-dimensional space is modelled by a Student-t distribution in computing the conditional probability between points [38]. Similarities between two points y_j and y_i in the learned low-dimensional space is modelled and shown in Equation 5.3.

5.7 Proposed Method

Our objective is to map emotion in the form of a label to a phrase which would then be used as an utterance by a virtual human in the prisoner's dilemma game. We adopt an approach that utilizes Word2Vec embeddings at the sentence level. As one of the main shortcomings of Word2Vec is capturing sentiment and emotion, we attempted to capture emotion in the mapped phrases. Another key consideration is the context of the game itself. In particular, depending on the outcome of a single round of the prisoner's dilemma game, it was important to capture context of the game in delivering a more natural and realistic utterance. Therefore, our goal was two-fold and is as follows.

Objectives

1. Construct a phrase mapping that affectively aligns to a given emotion.

2. Account for the context of the prisoner's dilemma game by capturing game context into the phrase.

Our approach first involves calculating the embedding of the emotion label using the pretrained model described in Section 5.5.2. Second we compute the sentence embedding of a phrase by averaging the embeddings of each word in the sentence after the removal of stop words. Stop words are removed using the stop word list provided by the NLTK[1] We query the appropriate phrase by constraining the query to a single bin, discussed below and return the phrase with the highest relative cosine similarity as defined in Section 4.15. With these considerations, we present how we accomplish our objectives outlined above.

To accomplish objective 1, we hand craft phrases based on the polarity of evaluation in the EPA vector of potential emotion labels. If considering given emotions with a positive evaluation we design phrases satisfying this constraint. To accomplish objective 2, we enumerate all move combinations between both agent and client. After taking both of these constraints into consideration we enumerate all possible combinations which we refer to as **bins**. As a result, in crafting and designing phrases we assign phrases to 1 of the 8 bins. When designing phrases, we first considered crafting phrases based on the game context. After, we filtered phrases into bins by rating phrases on their polarity of the evaluation score. The polarity of the phrase was computed using a sentiment analysis package. To determine the polarity of a phrase, we average the polarity of each word after removing stop words. Based on polarity of the evaluation, we assigned it to one of the 8 respective bins.

The complete enumeration of the 8 bins listed below.

Phrases are assigned to one of following **bins**:

1. Agent Give 2 and Client Give 2 and Evaluation < 0
2. Agent Give 2 and Client Give 2 and Evaluation > 0
3. Agent Give 2 and Client Take 1 and Evaluation < 0
4. Agent Give 2 and Client Take 1 and Evaluation > 0
5. Agent Take 1 and Client Take 1 and Evaluation < 0
6. Agent Take 1 and Client Take 1 and Evaluation > 0

[1] https://www.nltk.org/

7. Agent Take 1 and Client Give 2 and Evaluation < 0
8. Agent Take 1 and Client Give 2 and Evaluation > 0

Phrases within these **bins** span various emotions. Therefore, given an emotion, we query a phrase by first identifying the appropriate bin. Secondly, we compute the embedding of the emotion label using the pretrained Google model. Finally, we take the closest phrase resulting from the highest cosine similarity. It is our intention that these design considerations will enhance the mappings for game utterances in the prisoner's dilemma providing a more emotionally and contextually aligned utterance.

5.8 Phrases

We present an example of one bin containing the hand-crafted phrases used in this work. The full 8 bins containing the hand-crafted phrases used in this work is shown in Appendix C.

Give 2, Give 2 Positive Evaluation:

1. I appreciate that.
2. Thanks!
3. You are too nice.
4. This is fun.
5. I am having fun are you
6. Ok.
7. Great!
8. Thanks for being friendly
9. Thank you I am delighted
10. I am a happy person.
11. I like playing with you.
12. You are too generous.
13. This is looking good.
14. It is always good to have some perspective
15. do you know that I love this game?
16. did I tell you that I love this game?
17. It is a nice day today.
18. I love this!
19. I love this game!
20. This is going pretty well.
21. I can be sentimental at times.
22. I will try to be more civil
23. Well that was courteous of you!
24. Thank you very much.
25. This is very pleasant
26. I am often told that I am a lively person
27. I think I have a sparkling personality
28. At times it is important to be aggressive
29. I believe it is alright to be a hostile

person
30. I am beginning to respect you
31. I want to be respected
32. I am well respected amongst my peers
33. what can I say everyone admires me
34. Well aren't we both in a cheerful mood!
35. I am very sympathetic
36. See, I do care about you
37. I feel sorry for you
38. I think I may regret that later on
39. I am very thankful for that
40. Thanks I appreciate it
41. This is a satisfying game
42. I can accept that
43. Thanks I am pleased
44. Very pleasant very pleasant indeed
45. wow I am in awe
46. I am a little wonderstruck
47. I have to admit I am very anxious
48. I have to admit I am very optimistic
49. I am just full of hope
50. this game got me overjoyed
51. there you go I am sorry for you

5.8.1 Visualization of Phrases

We present a visualization for phrases from the bins with a positive evaluation mentioned in Section 5.8. We selected a subset of positive emotions from an ACT dictionary computed our mapping technique to phrases. We observe a reasonable mapping given the relatively small set of phrases constructed. One observation is that phrases mapped to "happy" do not seem to cluster accurately suggesting that "happy" may be indistinguishable from other emotions with a positive evaluation. As a result, the word embedding for "happy" can mapped to a variety of emotions with a positive evaluation.

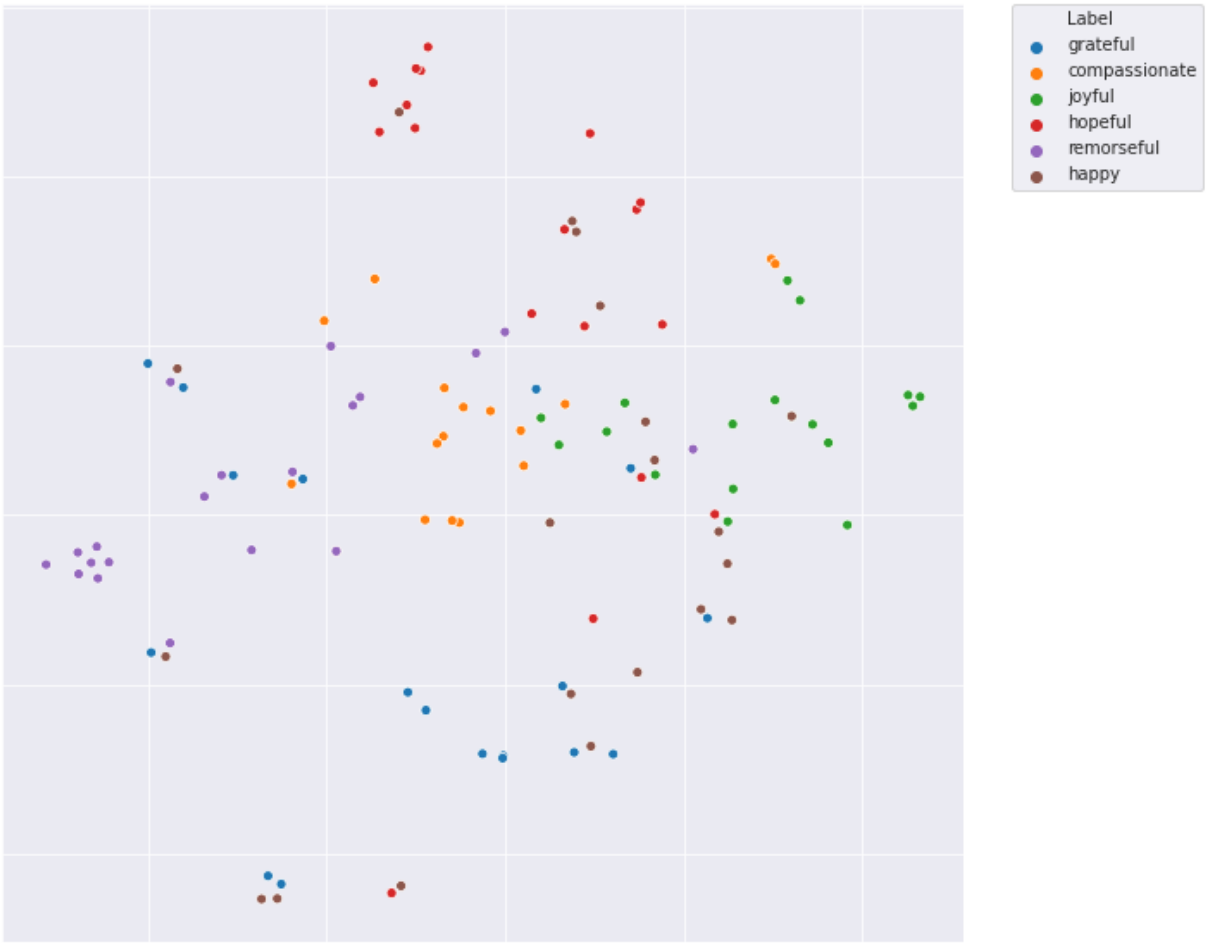


Figure 5.1: t-SNE visualization of phrases being mapped to bins with a positive evaluation bin.

e

Chapter 6

Prisoner's Dilemma Experiments

This chapter presents the methodology and results of our experiments exploring affect within the context of the prisoner's dilemma. We investigate interactions between human players and OCC agents. An OCC agent follows an appraisal based emotional model where actions are determined by mapping appraised emotions and game history to a set of actions for the agent to take. In a second experiment, we investigate interactions between human players and ACT agents, and interactions between human players and OCC agents. Our ACT agent follows Affect Control Theory (ACT) principles in which this agent aims to minimize the difference between the fundamental sentiments and transient impressions.

We present a proof of concept our work by demonstrating the feasibility of studying affective virtual agents within the context of the prisoner's dilemma in an experiment. Our work presented in Chapters 3,4 and C.1 is culminated in the following prisoner's dilemma experiments.

6.1 Prisoner's Dilemma Experiment I

6.1.1 Description of Experiment

We recruited participants on the Amazon mechanical turk platform. In total we recruited 124 participants (74 male, 48 female, 1 identified as other, and 1 did not wish to share). The age range of participants are between 12 and 74. In terms of remuneration, participants earned $0.70 plus an additional bonus of $0.50 per every point that they earned in the game. On Amazon mechanical turk we specified constraints in which only qualified participants

can participate in our experiment. These qualifications included a geographic restriction in which participants be from North America. Two additional qualifications are as follows. Only participants who completed 50 or more HITs and had earned an approval rating of 95% or more. All materials and details regarding the experiment has been approved by the University of Waterloo Office of Research Ethics.

6.1.2 Protocol and Procedure

These participants were required to sign a consent form and provided basic demographic information. Our experiment consists of two main components. That is, the prisoner's dilemma game followed by a questionnaire. As after signing into our application, participants were presented with instructions playing the prisoner's dilemma game as it pertains to our gaming interface. The instructions were delivered through verbal and non-verbal cues transmitted from a virtual human as discussed in Chapter 3. Participants were grouped into two groups based on a rearrangement of the questionnaires and then assigned one of four experimental conditions. Participants played 25 rounds but were told that they would play up to 30 rounds of the prisoner's dilemma game in an effort to mitigate irregular changes in strategies occurring near the end of the game. Given that the total potential bonus was significantly higher the base payment, our intention was that participants would be more attentive as they would attempt to maximize their points earned in the game. In our prisoner's dilemma game, the reward matrix reflects Table 2.1.

Participants were evenly divided into 3 agent groups. The virtual human transmitted two emotional signals through non-verbal and verbal cues. That is, speech and facial expressions. These methods are described in Chapters 4 and 5.

6.1.3 Experimental Conditions

1. OCC agent
2. Random agent
3. Emotionless agent

We will briefly describe the strategies and emotion models used in each experimental condition of our prisoner's dilemma game. In all experimental conditions the strategy implemented for the agent was tit-for-two-tat. In the tit-for-two tat strategy, a player only defects when their opponent defects twice in a row.

The model used for the OCC agent is discussed in Section 2.2.1. Based on the OCC model presented in Section 2.2.1, emotions are appraised on the consequences of events and the actions of agents. OCC emotional appraisals for the PD game are displayed in Appendix B.1. We utilize a set of coping rules which takes emotional appraisals and maps it to actions based on the game history. The coping rules are shown in Appendix B.3. Two of our agents serve as baselines where one is emotionless in the sense that it does not provide any verbal or non-verbal cues. The second is random, in the sense that the emotion selected is random, but still transmits facial expressions and utterances aligning with the randomly selected emotion. All agents except the emotionless agent utilize facial expression and utterance methods of which are consistent. For facial expressions, method I discussed in Chapter 4 is utilized.

6.1.4 Evaluating for Humanness

Studies of the attribution of humanness have described distinctive traits integral to human species as human uniqueness (HU) and properties that are fundamental to humans as human nature (HN) [33, 5]. As a result, humanness can be described in two dimensional space (HU, HN). Civility, refinement, maturity, rationality, and moral sensibility are the attributes that form the HU dimension. While attributes such as emotionality, warmth, openness, individuality, and depth form the HN dimension. HU distinguishes groups from animalistic traits such that groups not associated with HU would tend to exhibit more animalistic qualities. HN distinguishes humans from machines. Formally, the association to HU is formed by distinguishing between the following attributes.

- Civility vs. lack of culture
- Refinement vs. coarseness
- Moral Sensibility vs. amorality
- Rationality vs. irrationality
- Maturity vs. childlikeness

Similarly, the association to HN is formed by distinguishing between the following attributes.

- Emotional responsiveness vs. inertness

- Interpersonal warmth vs. coldness
- Cognitive openness vs. rigidity
- Individuality vs. passivity
- Depth vs. superficiality

6.1.5 Evaluation Overview

Based on the above formulation, we evaluate our experimental conditions (with facial expression and utterance modules) by determining where they lie in the two dimensional space based on the questionnaire results. In addition, we also analyze other statistics such as cooperation and defection rates under all experimental conditions. We hypothesize that there is a tendency to cooperate more with the agent that exhibits more humanisitic qualities. We ask participants to rate the traits listed above that form human uniqueness and human nature on a continuous scale ranging from [-4,4].

6.1.6 Results

In this section, we present our results for experiment I. We evaluate all agents based on humanness and cooperation.

Humanness

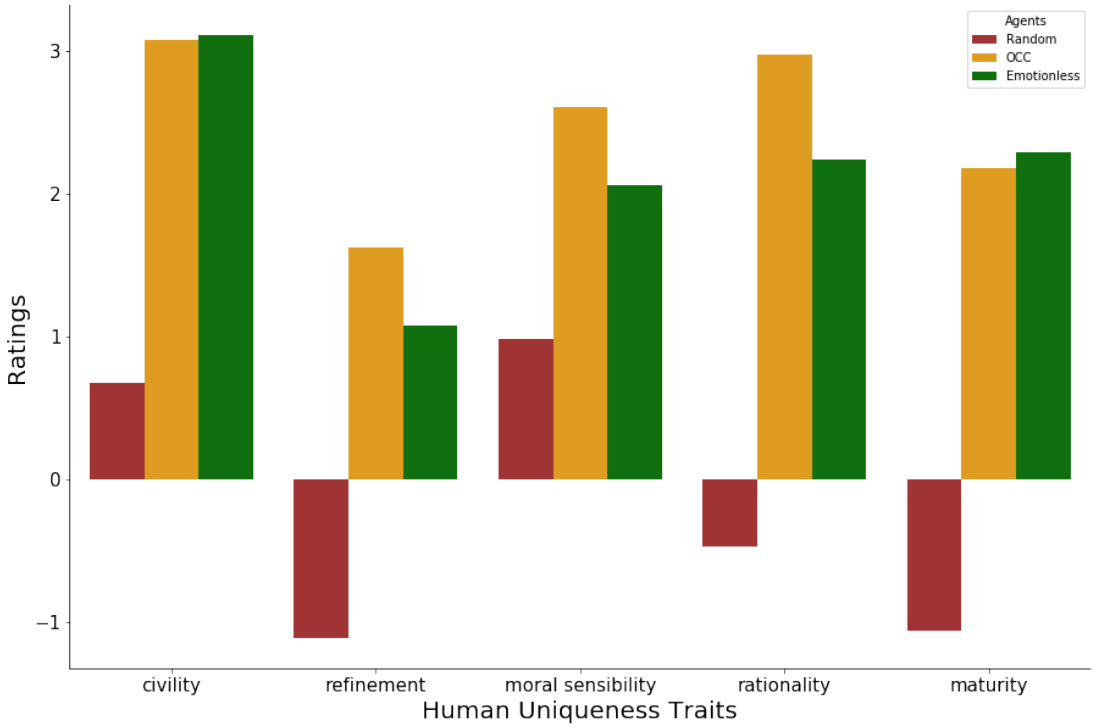


Figure 6.1: Ratings for the traits that make up human uniqueness. Positive values indicate a strong association to the human uniqueness traits while negative values have a strong association with animalistic qualities.

In Figure 6.1 we can observe that the OCC agent had positive ratings with human uniqueness attributes indicating a stronger association with humanistic qualities.

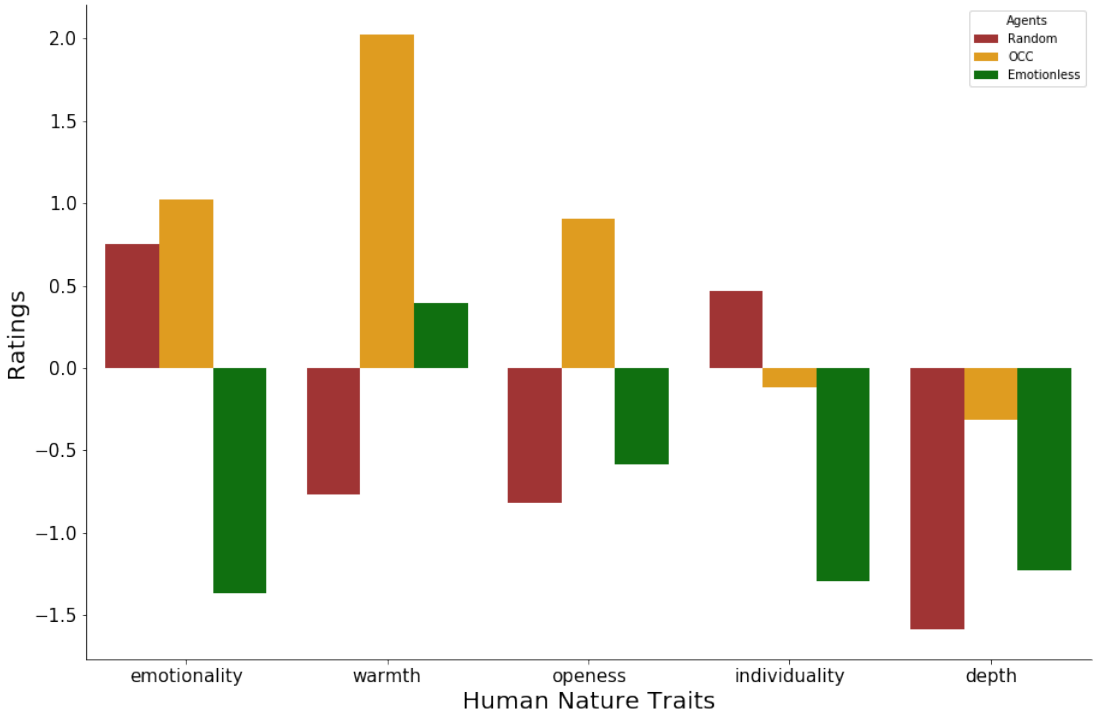


Figure 6.2: Ratings for the traits that make up human nature. Positive values indicate a stronger association to the human nature traits while negative values have a strong association with machine-like qualities.

Human nature ratings are displayed in Figure 6.2. These results suggest that OCC has a stronger association to all human nature attributes except for the individuality attribute. Although OCC performed poorly for ratings of the depth trait, it is clear that all agents were rated poorly for depth. As a result, these ratings for depth are more associated with superficiality as shown in Section 6.1.4.

We also asked participants to directly rate human-like versus machine-like and human-like versus animal-like shown in Figure 6.3. We believe these results are inconclusive as it is likely that participants associated the agent's appearance to these traits as opposed to the agent's emotion and behaviour.

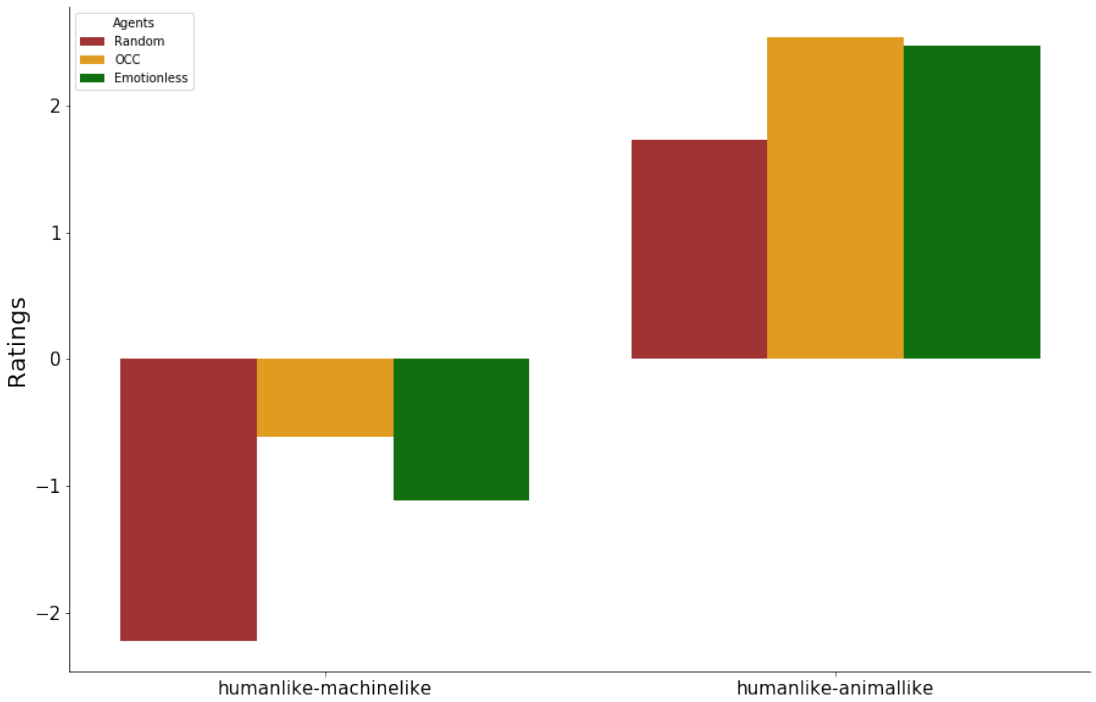


Figure 6.3: Ratings for human-like versus machine-like and human-like versus animal-like. Positive values are more strongly correlated to human-like.

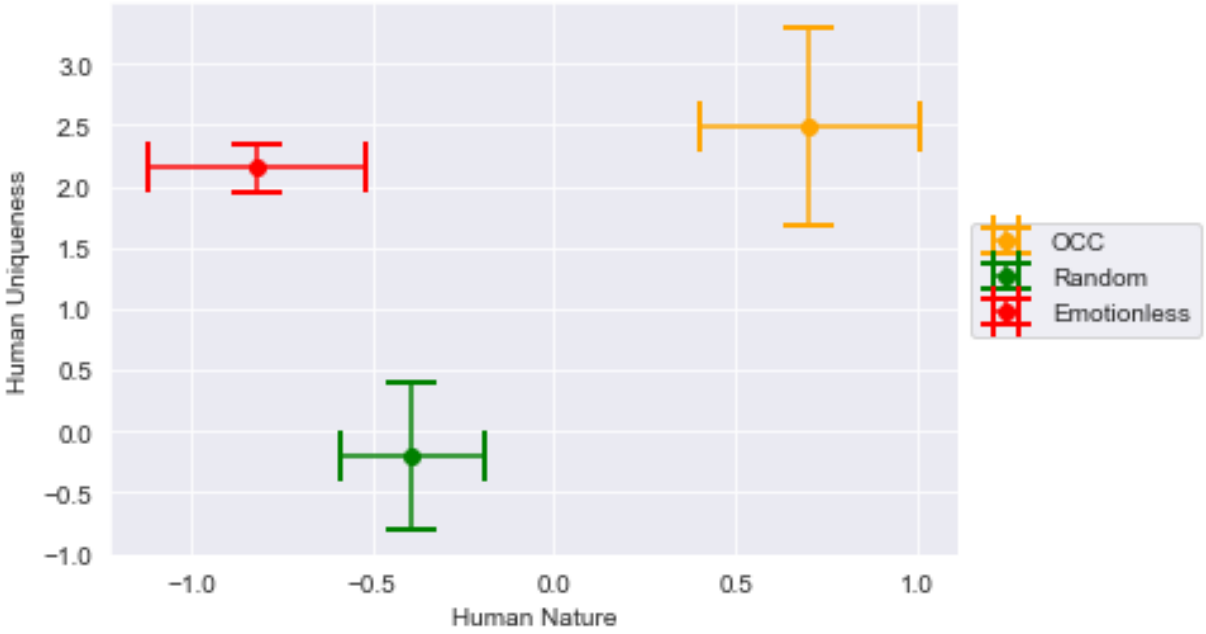


Figure 6.4: Humanness ratings in human uniqueness and human nature 2-dimensional space. 95% confidence intervals are displayed.

Figure 6.4 displays the ratings for all agents in human uniqueness and human nature dimensions by combining the attributes that form the 2-dimensional space. These results suggest that the OCC agent exhibits qualities that are associated to humanness with respect to both dimensions. As hypothesized, the random agent scores very poorly in humanness dimensional space. In addition, the emotionless agent scores poorly in the human nature dimension resulting in a stronger association to machine-like attributes. This suggests that the absence of emotional cues such as facial expressions and utterances in the emotionless condition resulted in a poor rating in attributing humanness.

Cooperation

We asked if different emotional displays affect the strategies of participants. As all agents played the same strategy of tit-for-two-tat we attribute the change in participants cooperation in all agents due to the effect of the emotional display. In Figure 6.5 the cooperation rate for participants are plotted for each round. We observe a higher and more stable consistent cooperation with participants who played against the OCC agent. Figure 6.6

shows the total number of rounds of cooperation for participants. Our results indicate statistical significance ($p < 0.05$) in the difference of cooperation between OCC agent and the random agent. The cooperation rates of between the OCC agent and emotionless agent show no statistical significance.Figure 6.7 shows the mean cooperation of the agents. As all agents played a tit-for-two-tat strategy we can observe that the agents cooperation reflects the participants cooperation.

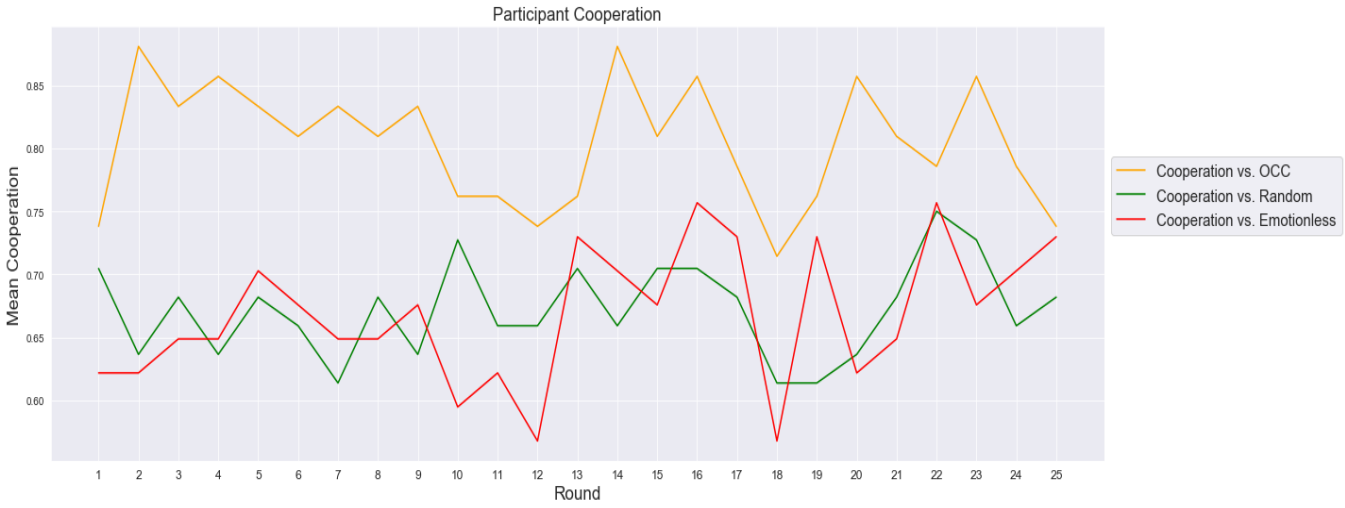


Figure 6.5: Mean participation cooperation rate over consecutive rounds against each agent.

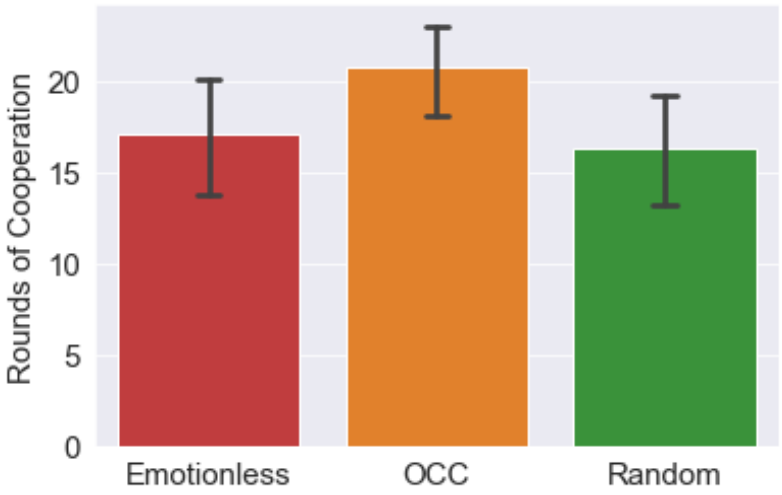


Figure 6.6: Total cooperation rounds for each agent with 95% confidence intervals.

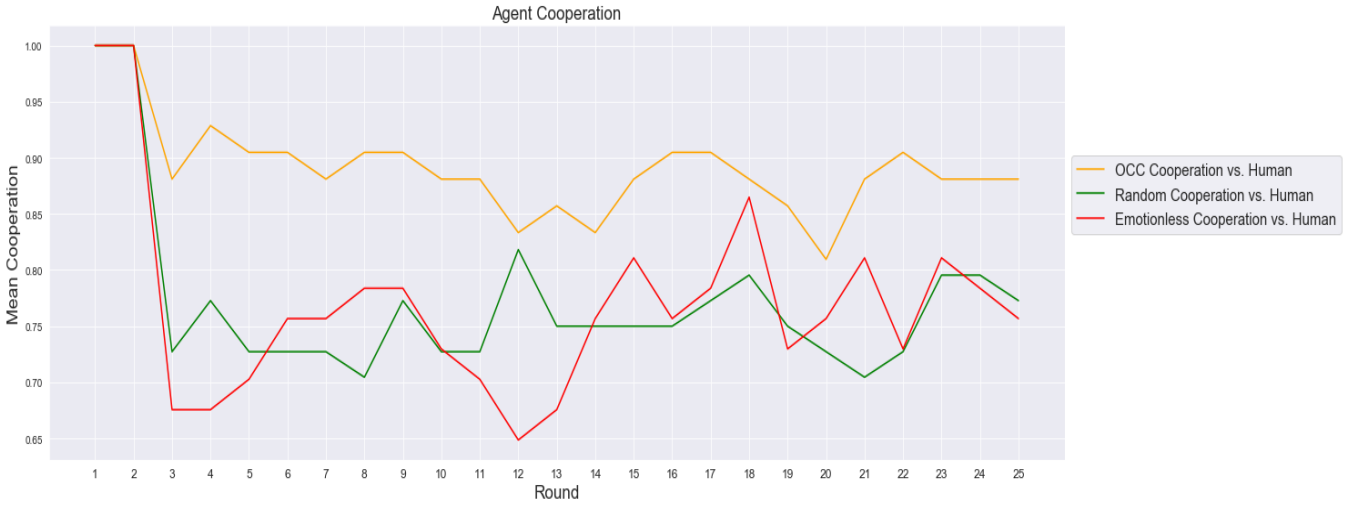


Figure 6.7: Mean agent cooperation rate over consecutive rounds against each agent.

6.2 Prisoner's Dilemma Experiment II

The experimental protocol and all other details remain the same as the previous experiment except for the following details. Below are the following experimental conditions used in

our second experiment.

6.2.1 Experimental Conditions

1. ACT agent
2. OCC agent
3. Random agent
4. Emotionless agent

We conduct a second experiment with the following modifications. We integrate an ACT agent as discussed in Section 2.4. The ACT agent does not require any coping mechanisms and instead actions are derived from the model. The ACT agent utilizes a larger set of emotions and and identities. We use our novel and improved method III described in Chapter 4 for the facial expression mappings for all experimental conditions except for the emotionless agent which does not display any non-verbal or verbal cues. We test out a different playing strategy. The strategy for all agents is a modified tit-for-tat. In a tit-for-tat strategy, the agent will cooperate initially and defect only when their opponent defects in the previous round. In our modified experiment, we experimented with having agents first defect and then follow the standard tit-for-tat strategy. Therefore, there is less of a tolerance for defection as compared to the previous tit-for-two-tat strategy. The OCC agent follows a modified coping mechanism which is defined in Appendix B.3. We also did not ask participants to directly rate human-like versus machine-like and human-like versus animal-like as results from experiment I were inconclusive based on the reasoning we presented above. Besides on these modifications described above, all other procedures and protocol remain the same as described in experiment I.

We recruited participants on the Amazon mechanical turk platform. In total we recruited 91 participants (56 male, 35 female). The age range of participants are between 19 to 64. In terms of remuneration, participants earned \$0.70 plus an additional bonus of \$0.50 per every point that they earned in the game. On Amazon mechanical turk we specified constraints in which only qualified participants can participate in our experiment. These qualifications included a geographic restriction in which participants be from North America. Two additional qualifications are as follows. Only participants who completed 50 or more HITs and had earned an approval rating of 95% or more. All materials and details regarding the experiment has been approved by the University of Waterloo Office of Research Ethics.

6.2.2 Results

In this section, we discuss our results based on the same empirical measures described in experiment I in Section 6.1.5.

Humanness

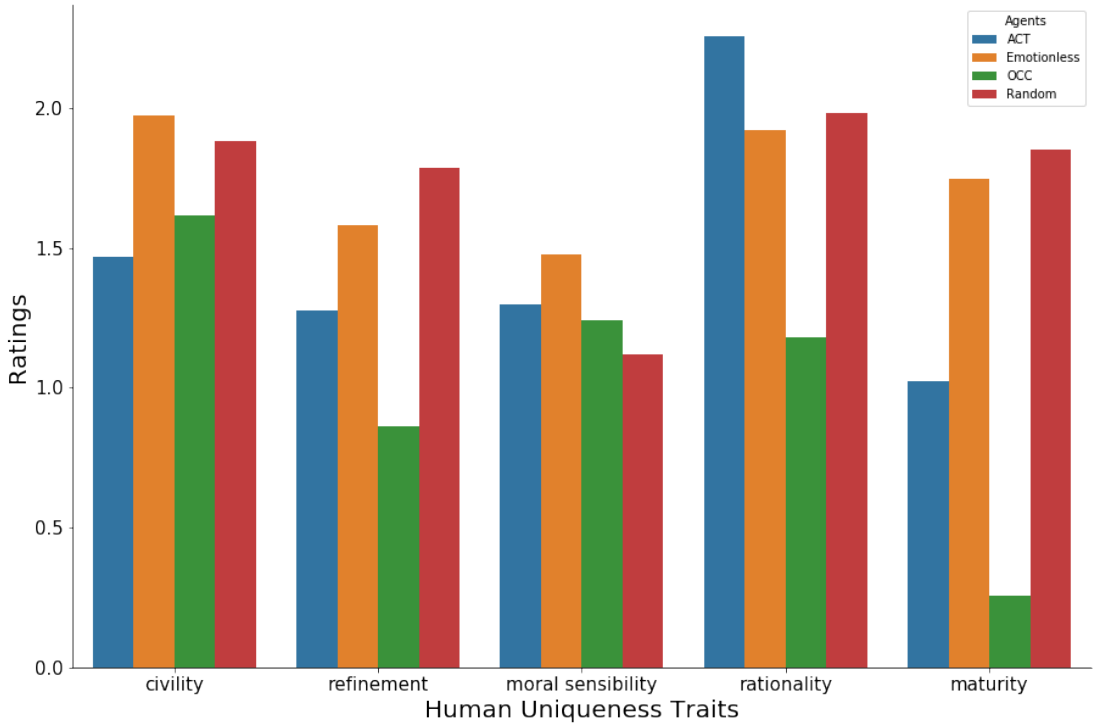


Figure 6.8: Ratings for the traits that make up human uniqueness. Positive values indicate a strong association to the human uniqueness traits while negative values have a strong association with animalistic qualities.

Figure 6.8 shows the ratings for the traits the form the human uniqueness dimension. We can observe that all agents have been rated positively for all traits meaning that they are more associated to humanistic attributes. However, it is evident that the ACT agent had only scored higher on the rationality dimension. These results indicate the the OCC agent has performed poorly on these human uniqueness traits. This may be a direct result of the modification of the coping mechanism. We believe that since the coping strategies

have been modified to make the OCC agent less cooperative this may have had a negative influence. Our results indicate that as our baseline conditions have been rated higher in human uniqueness traits (civility, refinement, moral sensibility, and maturity) than the OCC agent. We believe that the differences in humanness ratings for the OCC agent in the second experiment can be attributed to the fact that the agent defects more.

Figure 6.9 shows the human nature ratings for each trait forming this dimension. As also observed from experiment I, there seems to be a poor ratings for the depth trait which remains consistent with experiment I in the sense that the negative values indicate a weaker association to human nature. Nonetheless, we can observe that for the emotionless agent, depth has the lowest rating in comparison to the other agents. We observe that our baseline condition for the random agent has positive results between greater than 0.5 and less than 2 for emotionality, warmth and openness. We are unable to draw any conclusions supporting ACT and OCC agents for human nature traits. However, as expected the negative ratings for our emotionless agent indicate a stronger association to machine-like qualities. This can be explained based on the absence of non-verbal and verbal cues, suggesting that the facial expressions and utterances may be positively correlated with human-like attributes.

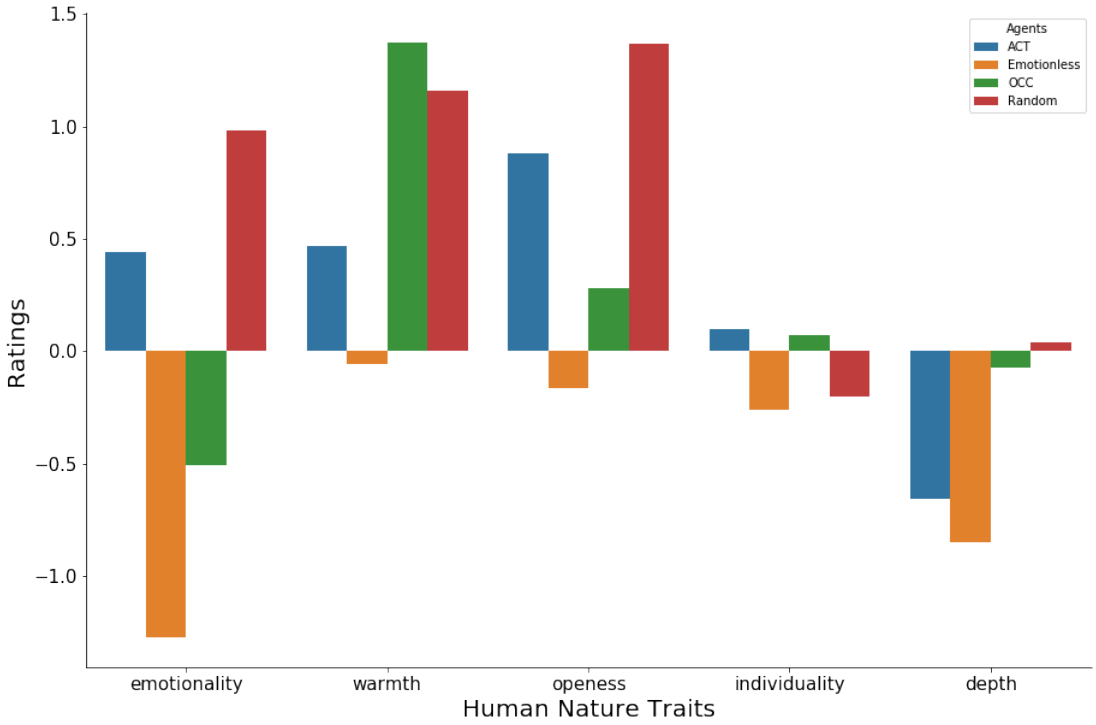


Figure 6.9: Ratings for the traits that make up human nature. Positive values indicate a stronger association to the human nature traits while negative values have a strong association with machine-like qualities.

Figure 6.10 shows the dimensional plot of the combined traits forming the human uniqueness and human nature dimensions. We are able to better visualize where each agent lies in the HU and HN space. We can observe that the random agent is rated positively for both human nature and human uniqueness dimensions. These results indicate that while the ACT agent is positively plotted in both HU and HN dimensions, the results indicate a stronger association to humanness for the random baseline condition. Our ACT agent is rated better in the combined space of HU and Hn indicating a stronger association to humanness. Based on the overlap in Figure 6.10, these differences are not significant.

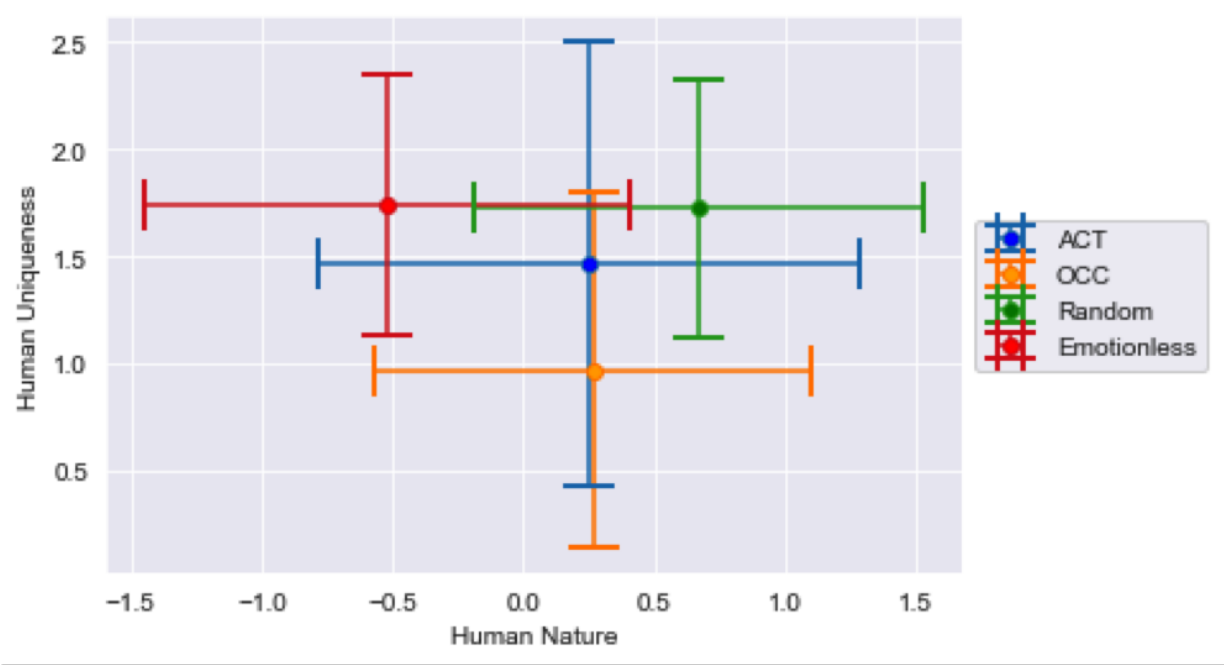


Figure 6.10: Humanness ratings in human uniqueness and human nature 2-dimensional space.

Cooperation

Similarly as in experiment I, we ask if different emotional displays affect the strategies of participants. As all agents played the same strategy of tit-for-tat we attribute the change in participants cooperation in all agents due to the effect of the emotional display. We observe a higher cooperation in both OCC and ACT agents than our baseline conditions of emotionless and random. Figure 6.11 shows the mean cooperation over consecutive rounds of the prisoner's dilemma game. It can be observed that there was more human cooperation with the ACT and OCC agents. Figure 6.12 shows the total cooperation in rounds for each experimental condition. While there is no statistical significance across each condition, we can see a total drop in the total number of cooperations from experiment I. This is explained as the change in strategy to tit-for-tat results in less of a tolerance in defection and as a result participants have responded in a less cooperating manner. Figure 6.13 shows the agents mean cooperation against human participants over consecutive rounds. As we can observe the ACT agent mean cooperation tends to remain at a higher level than the other conditions.

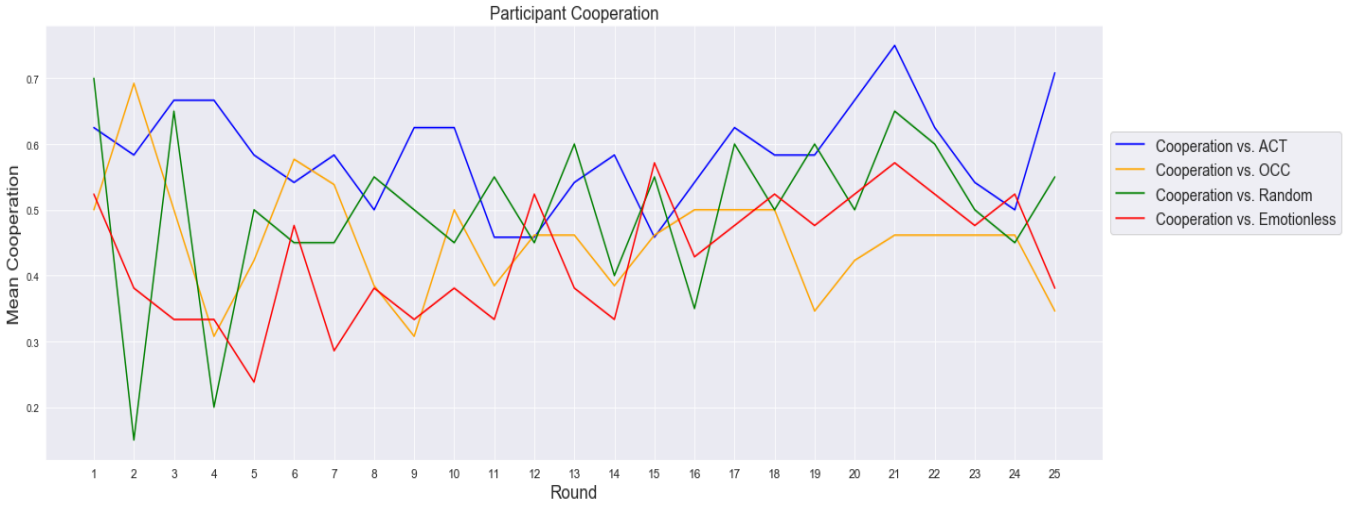


Figure 6.11: Mean participant cooperation rate over consecutive rounds against each agent.

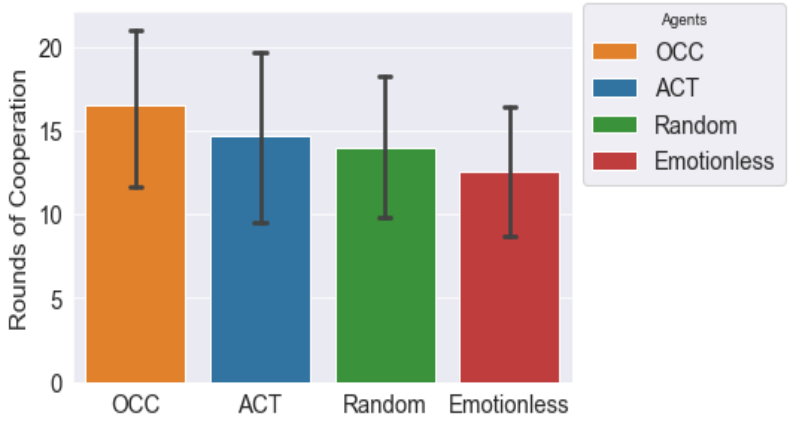


Figure 6.12: Total cooperation rounds for each agent with 95% confidence intervals.

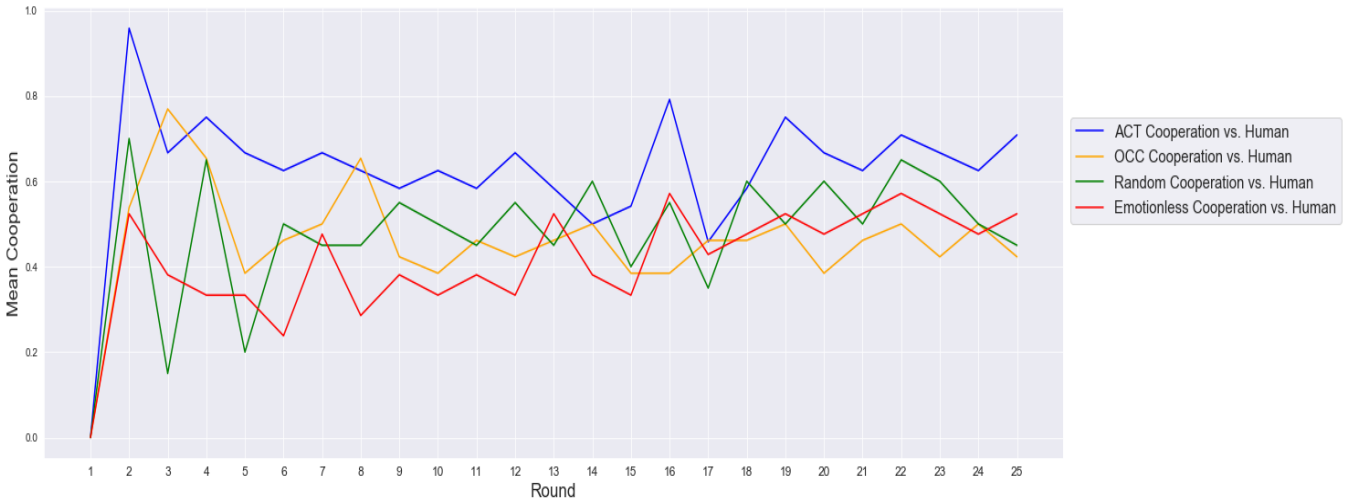


Figure 6.13: Mean agent cooperation rate over consecutive rounds against each agent.

6.3 Limitations

In this section, we present some limitations with respect to our experiments.

We do anticipate that for participants it may be difficult to rate our virtual human within the context of the prisoner's dilemma on these traits. One of our primary concerns is that although we specify in the instructions to rate the agent based on her behaviour, it is difficult to determine if participants rate agents based on the virtual human's visual appearance and aesthetics. As a result, this presents one the limitations of our experimental setup.

As discussed in Chapter 3, participants have to select an emoji to express their emotions. The limitation of using emojis is that we are only able to capture a discrete set of emotions. A better method would be to integrate automatic affect recognition based on facial expression recognition.

Chapter 7

Conclusion

In this work, we study emotionally inclined AI agents within the context of social dilemmas. One limitation of interactive AI systems is the focus of meeting functional requirements. As a result, intelligence is often achieved at only a machine level. The interdisciplinary field of Affective Computing addresses the lack of human and emotional intelligence by placing an emphasis on the design of AI systems that can recognize, interpret and simulate human-like behaviour and emotion. Many interactive AI systems are designed for deployment in complex social environments. Thus, building artificial agents that exhibit human-like behaviour and emotion is an integral part in achieving human-level intelligence. Social dilemmas are social situations that are ubiquitous in modern society. Social dilemmas provide a social environment where we can study the effects of interactions of affective artificial agents. In this thesis, we address this problem of studying human-like and affective virtual agents. Our research goals are two-fold. That is, our first aim is to present a proven and robust platform for studying emotionally cognizant AI agents within the prisoner's dilemma. Our work sets the foundation for research and studies into more complex and autonomous agents within the context of other social dilemmas. Our second aim is propose novel machine learning methods to map dimensional models of emotion to affective facial expressions and affective utterances for virtual human agents in the prisoner's dilemma. The contributions of our work presented in this dissertation is summarized as follows.

- In Chapter 3, we provided an overview of our prisoner's dilemma application. We design and implement a prisoner's dilemma game application with the integration of a virtual human. The integration of a virtual human agent provided functional mechanisms enabling the usage facial expressions and speech signals. We provide

an overview of game development principles and practices we have adopted in the implementation and design of our game.

- In Chapter 4, we propose three methods to map dimensional representations of affect to a virtual human facial configuration in HSF space. Our first method is uses linear algebraic principles to map EPA vectors to virtual human facial configurations in HSF space. For our second method, we study two state-of-the-art deep CNN architectures, VGG-16, and ResNet-50 used in face recognition tasks. We fine tune these models for affect recognition of valence and arousal dimensions. We experiment with transfer learning techniques, and determined that we could effectively transfer learning from the domain of face recognition to affect recognition. In our final method, we have presented a novel and robust approach to map dimensional representations of affect to virtual human facial configurations by modifying the VGG-16 architecture. We generate our own dataset of 9,200 virtual human facial images with their corresponding configurations in HSF space. We modify a VGG-16 (trained on VGGFace dataset) architecture in such a way that we can extract a feature vector of an input image, and thus generate a more compact representation of a facial image. This compact representation of an image is leveraged in the construction of a customized dataset from which we train a simple feed-forward network to predict the HSF space configurations directly. We observe through a few visual comparisons our third method can capture negative emotions at a finer level. That is, the facial expression construction of method III seems to show evidence suggesting a better representation for emotions with a negative evaluation, although more evaluation is necessary.

- In Chapter 5, we devise a method leveraging natural language processing techniques and distributing word representations to construct a mapping from emotion to text at the sentence level. We provide some insight into constructing phrases that capture with game context and affect.

- In Chapter 6, we show empirical evidence that agents that are capable of showing emotion will be perceived to being more human-like. More importantly, we show that our application can be used successfully in studying human-like AI agents within the context of the prisoner's dilemma game. Our empirical results indicate that an appraisal based theoretic agent is more human-like than baseline models.

7.1 Future Work

This work opens up many new research directions related to research in affective AI agents. We have designed and built an application that has successfully been used to study and evaluate agents with in the prisoner's dilemma. Our work presents a robust platform and stepping stone for future research into conducting further experiments involving the studying humanness of affective AI agents within the context of other social dilemmas. It would be interesting to study affectively inclined AI agents within the context of other social dilemmas. This includes the assurance game and chicken game, described briefly in Chapter 2.

We now address some limitations and some additional areas for future work. Dimensional models have proven to capture more subtle emotion variations and can differentiate between different emotion intensities. However, while dimensional models of emotion presents a more finer representation of human emotion and behaviour, it presents some limitations. One of the main limitations stems from the inaccuracies of capturing the denotative meaning of words. For instance, consider the meaning of the two words "repentant" and "reverent". In terms of an interpretation of these words, they have different meanings and as a result, should result in different facial expression. However, in EPA space these words have almost identical EPA vectors. Consider the EPAs of repentant, [1.5714, 1.3143, -0.8714] and reverent = [1.5724, 1.1517, -0.8414] [1]. It is apparent that these EPAs lie closely in 3-dimensional vector space. As a result, a similar facial expression would be constructed for these affectively different emotions. Therefore, we strongly believe that more powerful representations of emotions are necessary. Representations that can capture both a deeper semantic relationship in text with respect to their denotative meaning is one potential area of future work. More powerful and accurate representations of emotions would result in only more accurate algorithms used in mapping emotions to text and facial expressions. In this work, we also observe the lack of versatility of the utterances used in Chapter 5. It may be interesting to experiment with generative models that can generate a response word-by-word, given some input signal in the form of a query.

Another, limitation is the lack of facial images in areas of the valence and arousal space containing less samples to draw from. One potential area for future work is to generate synthetic samples in an effort to balance the database. Generative Adversarial Networks (GAN) can be been used to address this limitation [29]. A GAN is a deep neural network architecture that consists of two adversarial models, one of which is a generator and the second of which is a discriminator. Intuitively, the idea of a GAN is to have the generator

[1] All of these EPA values are taken from the Indiana 2002-2004 dataset [37].

take random noise and generate a "fake" output. This "fake" output and the real output is passed to the discriminator. The discriminator will determine if the generated output belongs to the class of the real output. The aim a GAN is to generate samples that can eventually "fool" the discriminator. Therefore, usage of GANs may also be of potential interest in generating synthetic images of our virtual human dataset and as well as real images.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Carole Adam, Andreas Herzig, and Dominique Longin. A logical formalization of the occ theory of emotions. *Synthese*, 168(2):201–248, 2009.

[3] Martin Adam, Michael Wessel, and Alexander Benlian. Ai-based chatbots in customer service and their effects on user compliance. *Electronic Markets*, pages 1–19, 2020.

[4] Karl J Astrom. Optimal control of markov decision processes with incomplete state estimation. *J. Math. Anal. Applic.*, 10:174–205, 1965.

[5] Paul Bain, Joonha Park, Christopher Kwok, and Nick Haslam. Attributing human uniqueness and human nature to cultural groups: Distinct forms of subtle dehumanization. *Group Processes & Intergroup Relations*, 12(6):789–805, 2009.

[6] Debasish Basak, Srimanta Pal, and Dipak Patranabis. Support vector regression. *Neural Information Processing Letters and Reviews*, 11, 11 2007.

[7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[8] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[9] Ray L Birdwhistell. *Kinesics and context: Essays on body motion communication.* University of Pennsylvania press, 2010.

[10] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[11] Ronald Cole, Sarel Van Vuuren, Bryan Pellom, Kadri Hacioglu, Jiyong Ma, Javier Movellan, Scott Schwartz, David Wade-Stein, Wayne Ward, and Jie Yan. Perceptive animated interfaces: First steps toward a new paradigm for human-computer interaction. *Proceedings of the IEEE*, 91(9):1391–1405, 2003.

[12] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[14] Robert Dale. The return of the chatbots. *Natural Language Engineering*, 22(5):811–817, 2016.

[15] Celso M De Melo, Peter Carnevale, and Jonathan Gratch. The influence of emotions in embodied agents on human decision-making. In *International Conference on Intelligent Virtual Agents*, pages 357–370. Springer, 2010.

[16] Celso M De Melo, Peter Carnevale, and Jonathan Gratch. The impact of emotion displays in embodied agents on emergence of cooperation with people. *Presence: teleoperators and virtual environments*, 20(5):449–465, 2011.

[17] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[18] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112. IEEE, 2011.

[19] E Douglas-Cowie, R Cowie, I Sneddon, C Cox, L Lowry, M McRorie, L Jean-Claude Martin, JC Devillers, A Abrilian, S Batliner, et al. The humaine database: addressing

the needs of the affective computing community. In *Proc. of the Second Int. Conf. on Affective Computing and Intelligent Interaction*, pages 488–500, 2007.

[20] Paul Ekman. Are there basic emotions? 1992.

[21] Travis Faas. *An Introduction to HTML5 Game Development with Phaser. js.* CRC Press, 2017.

[22] C Fabian Benitez-Quiroz, Ramprakash Srinivasan, and Aleix M Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5562–5570, 2016.

[23] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[24] Wallace V Friesen, Paul Ekman, et al. Emfacs-7: Emotional facial action coding system. *Unpublished manuscript, University of California at San Francisco*, 2(36):1, 1983.

[25] Nico H Frijda et al. *The emotions*. Cambridge University Press, 1986.

[26] Moojan Ghafurian, Neil Budnarain, and Jesse Hoey. Role of emotions in perception of humanness of virtual agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1979–1981, 2019.

[27] Ulrich Gnewuch, Stefan Morana, and Alexander Maedche. Towards designing cooperative and social conversational agents for customer service. 12 2017.

[28] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[30] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, pages 117–124. Springer, 2013.

[31] Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269 – 306, 2004.

[32] Jonathan Gratch, Ning Wang, Anna Okhmatovskaia, Francois Lamothe, Mathieu Morales, Rick J van der Werf, and Louis-Philippe Morency. Can virtual humans be more engaging than real ones? In *International Conference on Human-Computer Interaction*, pages 286–297. Springer, 2007.

[33] Nick Haslam, Stephen Loughnan, Yoshihisa Kashima, and Paul Bain. Attributing and denying humanness to others. *European review of social psychology*, 19(1):55–85, 2008.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[35] Marti Hearst, S.T. Dumais, E. Osman, John Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13:18 – 28, 08 1998.

[36] David R Heise. *Expressive order: Confirming sentiments in social actions*. Springer Science & Business Media, 2007.

[37] David R Heise. *Surveying cultures: Discovering shared conceptions and sentiments*. John Wiley & Sons, 2010.

[38] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[39] J. Hoey, T. Schroder, and A. Alhothali. Bayesian affect control theory. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 166–172, 2013.

[40] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[41] Joshua DA Jung, Jesse Hoey, Jonathan H Morgan, Tobias Schröder, and Ingo Wolf. Grounding social interaction with affective intelligence. In *Canadian Conference on Artificial Intelligence*, pages 52–57. Springer, 2016.

[42] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on*

Automatic Face and Gesture Recognition (Cat. No. PR00580), pages 46–53. IEEE, 2000.

[43] Hyun Duk Kim, Dae Hoon Park, Yue Lu, and ChengXiang Zhai. Enriching text representation with frequent pattern mining for probabilistic topic modeling. *Proceedings of the American Society for Information Science and Technology*, 49(1):1–10, 2012.

[44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] Dimitrios Kollias, Panagiotis Tzirakis, Mihalis A Nicolaou, Athanasios Papaioannou, Guoying Zhao, Björn Schuller, Irene Kotsia, and Stefanos Zafeiriou. Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision*, 127(6-7):907–929, 2019.

[46] Dimitrios Kollias and Stefanos Zafeiriou. Aff-wild2: Extending the aff-wild database for affect recognition. *arXiv preprint arXiv:1811.07770*, 2018.

[47] Peter Kollock. Social dilemmas: The anatomy of cooperation. *Annual review of sociology*, 24(1):183–214, 1998.

[48] Jean Kossaifi, Georgios Tzimiropoulos, Sinisa Todorovic, and Maja Pantic. Afew-va database for valence and arousal estimation in-the-wild. *Image and Vision Computing*, 65:23–36, 2017.

[49] Peter J Lang, Margaret M Bradley, and Bruce N Cuthbert. Emotion, attention, and the startle reflex. *Psychological review*, 97(3):377, 1990.

[50] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[51] Charles Leifer. ORM, 2020.

[52] Luyuan Lin, Stephen Czarnuch, Aarti Malhotra, Lifei Yu, Tobias Schroeder, and Jesse Hoey. Affectively aligned cognitive assistance using bayesian affect control theory, 2014.

[53] Christine L Lisetti and Diane J Schiano. Automatic facial expression interpretation: Where human-computer interaction, artificial intelligence and cognitive science intersect. *Pragmatics & cognition*, 8(1):185–235, 2000.

[54] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[55] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, pages 94–101. IEEE, 2010.

[56] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. In *Proceedings Third IEEE international conference on automatic face and gesture recognition*, pages 200–205. IEEE, 1998.

[57] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[58] Lori Malatesta, Kostas Karpouzis, and Amaryllis Raouzaiou. Affective intelligence: the human face of ai. In *Artificial Intelligence An International Perspective*, pages 53–70. Springer, 2009.

[59] Aarti Malhotra, Jesse Hoey, Alexandra König, and Sarel van Vuuren. A study of elderly peoples emotional understanding of prompts given by virtual humans. In *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth 16, page 1316, Brussels, BEL, 2016. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[60] Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE transactions on affective computing*, 3(1):5–17, 2011.

[61] Roger Lee Mendoza. The hare question in assurance games: Practical problems and insights from robotic surgery. *The American Economist*, 63(1):18–30, 2018.

[62] László Mérő. *The Prisoner's Dilemma*, pages 28–47. Springer New York, New York, NY, 1998.

[63] Alex Mihailidis, Jennifer N. Boger, Marcelle Candido, and Jesse Hoey. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC Geriatrics*, 8(28), 2008.

[64] Alex Mihailidis, Jennifer N Boger, Tammy Craig, and Jesse Hoey. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC geriatrics*, 8(1):28, 2008.

[65] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[66] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017.

[67] Nils J Nilsson and Nils Johan Nilsson. *Artificial intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[68] Andrew Ortony, Gerald Clore, and Allan Collins. *The Cognitive Structure of Emotion*, volume 18. 01 1988.

[69] Charles E Osgood. The nature and measurement of meaning. *Psychological bulletin*, 49(3):197, 1952.

[70] Charles Egerton Osgood, William H May, Murray Samuel Miron, and Murray S Miron. *Cross-cultural universals of affective meaning*, volume 1. University of Illinois Press, 1975.

[71] Mike Owens. *The definitive guide to SQLite*. Apress, 2006.

[72] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *2005 IEEE international conference on multimedia and Expo*, pages 5–pp. IEEE, 2005.

[73] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[74] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[75] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[76] John Platt. Social traps. *American psychologist*, 28(8):641, 1973.

[77] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.

[78] Andrew B Raij, Kyle Johnsen, Robert F Dickerson, Benjamin C Lok, Marc S Cohen, Margaret Duerson, Rebecca Rainer Pauly, Amy O Stevens, Peggy Wagner, and D Scott Lind. Comparing interpersonal interactions with a virtual human to those with a real human. *IEEE transactions on visualization and computer graphics*, 13(3):443–457, 2007.

[79] Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. Introducing the recola multimodal corpus of remote collaborative and affective interactions. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–8. IEEE, 2013.

[80] Nick Abou Risk and Duane Szafron. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 159–166. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[81] Ira J Roseman. Cognitive determinants of emotion: A structural theory. *Review of personality & social psychology*, 1984.

[82] Ira J Roseman and Craig A Smith. Appraisal theory. *Appraisal processes in emotion: Theory, methods, research*, pages 3–19, 2001.

[83] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[84] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In *European workshop on biometrics and identity management*, pages 47–56. Springer, 2008.

[85] Klaus R Scherer. Studying the emotion-antecedent appraisal process: An expert system approach. *Cognition & Emotion*, 7(3-4):325–355, 1993.

[86] Klaus R Scherer, Angela Schorr, and Tom Johnstone. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, 2001.

[87] Tobias Schröder and Wolfgang Scholl. Affective dynamics of leadership: An experimental test of affect control theory. *Social Psychology Quarterly*, 72(2):180–197, 2009.

[88] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[89] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[90] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.

[91] Shimpei Soda, Masahide Nakamura, Shinsuke Matsumoto, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto. Implementing virtual agent as an interface for smart home voice control. In *2012 19th Asia-Pacific Software Engineering Conference*, volume 1, pages 342–345. IEEE, 2012.

[92] K Soumya George and Shibily Joseph. Text classification by augmenting bag of words (bow) representation with co-occurrence feature. *IOSR J. Comput. Eng*, 16(1):34–38, 2014.

[93] Photon Storm. Phaser HTML5 Game Framework, 2020.

[94] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

[95] Jianhua Tao and Tieniu Tan. Affective computing: A review. In *International Conference on Affective computing and intelligent interaction*, pages 981–995. Springer, 2005.

[96] Alaa Tharwat. Principal component analysis-a tutorial. *International Journal of Applied Pattern Recognition*, 3(3):197–240, 2016.

[97] Kristinn R. Thórisson and Justine Cassell. Why put an agent in a human body: The importance of communicative feedback in human-humanoid dialogu. 1996.

[98] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[99] A. W. Tucker. The mathematics of tucker: A sampler. *The Two-Year College Mathematics Journal*, 14(3):228–232, 1983.

[100] Sarel Van Vuuren and Leora R Cherney. A virtual therapist for speech and language therapy. In *International conference on intelligent virtual agents*, pages 438–448. Springer, 2014.

[101] Sarel van Vuuren and Leora R. Cherney. A virtual therapist for speech and language therapy. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Intelligent Virtual Agents*, pages 438–448, Cham, 2014. Springer International Publishing.

[102] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[103] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006.

[104] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[105] Stefanos Zafeiriou, Athanasios Papaioannou, Irene Kotsia, Mihalis Nicolaou, and Guoying Zhao. Facial affect"in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–47, 2016.

APPENDICES

Appendix A

Facial Expression Generation Comparison

A.1 Comparisons between Methods I, II, and III

Method I in column 1, Method II in column 2, and Method III in column III. The emotion is listed on the left

Disappointed



Hopeful



Relieved



Distressed




Remorseful
Angry
Smug
Happy

Resentful



Fearful



Appendix B

OCC Agent Details

B.1 OCC Agent Emotional Appraisals

The following Tables B.1 and B.3 are taken from [26].

B.2 Coping Rules for Experiment 1

We use five coping strategies as specified from [31]: acceptance, seeking support, restraint, growth, and denial. Upon game initialization the agent's hope leads to the support seeking coping mechanism, and thus to cooperation.

B.3 Coping Rules for Experiment 2

We use five coping strategies as specified from [31]: acceptance, seeking support, restraint, growth, and denial. Upon game initialization the agent's hope leads to the support seeking coping mechanism, and thus to cooperation.

Table B.1: OCC-based emotional appraisals in the PD game. The "consequences" and "actions of agents" correspond to the OCC decision tree. 🙂 means "pleased", 👍 means approving. ♡ means desirable, and ✓means confirmed. Aria is ambivalent for all lines not shown. For example, in the case where Aria gives while the player takes but shows regret, Aria does not disapprove of the player's action anymore (because he is showing regret), but does not actually approve of it either, so sits on the fence and does not feel admiration or reproach.

GAME PLAY				**VALENCED APPRAISALS**							**APPRAISED EMOTIONS**		
				Consequences						Actions of agents			
				other		self							
Previous	Most Recent					prospects relevant?							
	Move		Emotion		♡	yes			no			Momentary		Prospect-Based
						🙂?								
Player	Aria	Player	Player	🙂?	for other?	future	present	✓	🙂?	self 👍?	other 👍?	Single	Compound	Single
give 2	give 2	give 2	any	yes	yes	yes	yes	yes	yes	yes	yes	happy-for joy,pride joy,admiration	gratification gratitude	hope satisfaction
take 1	give 2	give 2	any	yes	yes	yes	yes	no	yes	yes	yes	happy-for joy,pride joy,admiration	gratification gratitude	hope relief
give 2	take 1	give 2	positive	no	no	yes	yes	yes	yes	no	yes	pity joy,admiration shame	gratitude	hope satisfaction
take 1	take 1	give 2	positive	no	no	yes	yes	yes	yes	no	yes	pity joy,admiration shame	gratitude	hope relief
give 2	take 1	give 2	negative	yes	no	no	yes	yes	yes	yes		gloating pride, joy	gratification	fear satisfaction
take 1	take 1	give 2	negative	yes	no	no	yes	no	yes	yes		gloating pride,joy	gratification	fear relief
give 2	give 2	take 1	no regret	no	yes	no	no	no	no	yes	no	resentment distress,reproach pride	anger	fear disappointment
take 1	give 2	take 1	no regret	no	yes	yes	no	yes	no	yes	no	resentment distress,reproach pride	anger	fear fears-confirmed
give 2	give 2	take 1	regret	no	yes	yes	no	no	no	yes		resentment distress pride		hope disappointment
take 1	give 2	take 1	regret	no	yes	yes	no	yes	no	yes		resentment distress pride		hope fears-confirmed
take 1	take 1	take 1	any	no	no	no	no	yes	no no	no	no	pity distress,shame distress,reproach	remorse anger	fear fears-confirmed
give 2	take 1	take 1	any	no	no	no	no	no	no no	no	no	pity distress,shame distress,reproach	remorse anger	fear disappointment

Table B.2: Experiment 1 Coping strategies for the OCC PD bot, including last player emotion, and the last two player moves.

Player moves		Player	coping		next
t-2	t-1	emotion (t-1)	strategy	example	move
take 1	take 1	-	acceptance: live with bad outcome	*"oh well, we're doomed"*	take 1
take 1	give 2	positive	growth: positive reinterpretation	*"this might be turning around"*	give 2
take 1	give 2	negative	growth+denial: positive reinterpretation	*"maybe he didn't mean that emotion"*	give 2
give 2	take 1	regret	restraint: hold back negative, keep trying	*"he's a good person really"*	give 2
give 2	take 1	not regret	denial: deny reality, continue to believe	*"maybe its not so bad"*	give 2
give 2	give 2	-	seek support: understanding and sympathy	*"Let's cooperate together on this"*	give 2

Table B.3: Experiment 2 Coping strategies for the OCC PD bot, including last player emotion, and the last two player moves.

Player moves		Player	coping	next
t-2	t-1	emotion (t-1)	strategy	move
take 1	take 1	-	acceptance: live with bad outcome	take 1
take 1	give 2	positive	growth: positive reinterpretation	give 2
take 1	give 2	negative	acceptance: positive reinterpretation	take 1
give 2	take 1	negative/regret	restraint+denial: hold back negative, keep trying	give 2
give 2	take 1	positive/not regret	acceptance deny reality, continue to believe	take 1
give 2	give 2	-	seek support: understanding and sympathy	give 2

Appendix C

Phrase listing

C.1 Complete Phrase List

Give 2, Give 2 Positive Evaluation:

1. I appreciate that.
2. Thanks!
3. You are too nice.
4. This is fun.
5. I am having fun are you
6. Ok.
7. Great!
8. Thanks for being friendly
9. Thank you I am delighted
10. I am a happy person.
11. I like playing with you.
12. You are too generous.
13. This is looking good.
14. It is always good to have some perspective
15. do you know that I love this game?
16. did I tell you that I love this game?
17. It is a nice day today.
18. I love this!
19. I love this game!
20. This is going pretty well.
21. I can be sentimental at times.
22. I will try to be more civil
23. Well that was courteous of you!
24. Thank you very much.
25. This is very pleasant
26. I am often told that I am a lively person
27. I think I have a sparkling personality
28. At times it is important to be aggressive
29. I believe it is alright to be a hostile person
30. I am beginning to respect you
31. I want to be respected
32. I am well respected amongst my peers
33. what can I say everyone admires me
34. Well aren't we both in a cheerful mood!

35. I am very sympathetic
36. See, I do care about you
37. I feel sorry for you
38. I think I may regret that later on
39. I am very thankful for that
40. Thanks I appreciate it
41. This is a satisfying game
42. I can accept that
43. Thanks I am pleased
44. Very pleasant very pleasant indeed
45. wow I am in awe
46. I am a little wonderstruck
47. I have to admit I am very anxious
48. I have to admit I am very optimistic
49. I am just full of hope
50. this game got me overjoyed
51. there you go I am sorry for you

Give 2, Give 2 Negative Evaluation:

1. I still hate playing this game
2. I still dislike you
3. I just hate you
4. Sometimes I may seem capricious
5. I know I may be capricious at times
6. I am still a little discouraged
7. I may still lose
8. I might lose
9. I don't know why I am still a bit irritated
10. That was dissatisfying
11. I think I may regret that
12. I am very proud of myself
13. Thanks I know I seem supercilious
14. I am just a angry person
15. I think I need to take a few deep breathes because I get angry easily
16. your welcome I was feeling a little embarrassed
17. there you go I am sorry for you
18. for some reason I feel sad
19. This game sometimes gets me feeling depressed
20. I am a little bummed I should have not done that
21. That was close I was very scared
22. I was worried for a bit
23. I was afraid for no reason

Give 2, Take 1 Positive Evaluation:

1. I am a joyful person.
2. You are very welcome
3. That was surprising!
4. well that was surprising.
5. I am very surprised.
6. I think I am being sensible
7. I can be sentimental at times.
8. I think I am the only cheerful one here
9. at least I am being sympathetic
10. why am I being nice
11. I am only being kind
12. I am still optimistic
13. It could have been alot worse
14. I am trying to be understanding

15. I am sort of a understanding person
16. at least I am being kind
17. I am still very confident
18. I am still thankful
19. I always try to be appreciative regardless of the outcome

Give 2, Take 1 Negative Evaluation:

1. That was outrageous
2. Your tactics were cunning
3. This is outrageous
4. You are making me upset.
5. That was mean
6. You are a terrible person.
7. You will regret that.
8. That was ridiculous.
9. That was very cruel
10. I regret that.
11. You are getting on my nerves.
12. You are being unfriendly.
13. Please do not make me agitated
14. Are you upset
15. I am disappointed.
16. You are making me frustrated
17. You are terrible
18. This is sad.
19. You are making me sad.
20. That was disappointing
21. I am so frustrated.
22. I feel like a fool.
23. I am about to panic
24. I am feeling a little agitated!
25. Do not panic!
26. I am very discouraged
27. This is getting really irritating.
28. This is looking bad
29. I am starting to get angry.
30. well that was daring of you!
31. well that was awkward
32. well that was rude of you
33. I am getting a little annoyed
34. you are getting on my nerves
35. I am feeling very irritated
36. You are irritating me
37. stop irritating me
38. I am starting to feel scared
39. I am a little afraid
40. I am starting to feel embarrassed

Take 1, Give 2 Positive Evaluation:

1. I never give up
2. I appreciate that
3. I am very sorry
4. I apologize
5. This is fun
6. Great!
7. Sorry about that.
8. Thank you I am delighted
9. This is looking good.
10. You know I am very discerning
11. Better luck next time.
12. do you know that I love this game?
13. did I tell you that I love this game?
14. I can be sentimental at times.
15. I think I am being sensible
16. I will try to be more civil
17. That was civil
18. Well that was courteous of you!
19. Thank you very much.
20. I don't know if I can trust you
21. I am enthralled about what just happened

22. very surprising, very surprising indeed
23. I will try to be more forbearing
24. I will try to be more patient
25. I think I am being forthright
26. In all honesty I am trying to win
27. I still have confidence in myself
28. I may be a little too confident
29. I might be greedy, but I am trying to win here
30. I know I am cunning
31. This is just me being conscientious
32. I am thrilled
33. Well wasn't that exhilarating
34. strange, very strange
35. strange, wasn't expecting that
36. This is good for me
37. I am doing very good
38. I am feeling a little blue!
39. very pleased at the outcome there
40. pleased to see you cooperate

Take 1, Give 2 Negative Evaluation:

1. Sometimes I need to be cruel
2. I feel terrible
3. I am starting to get angry
4. I think I may be feeling a little resentful
5. I did that because I was feeling disdainful
6. I apologize
7. I am sorry
8. I know I am a mean person sometimes
9. I am very sorry
10. I regret doing this to you but I want to win
11. I sense a bit of remorse
12. This game is sometimes irritating
13. I am scared to see what you are going to do to me next
14. sorry I was afraid
15. you have made me upset
16. I am extremely agitated at the moment
17. it is important to be aggressive
18. I am superior
19. I know I am slightly shameful.
20. this is a bit awkward
21. trust me I am embarrassed
22. believe me I am very embarrassed
23. This game gives me anxiety attacks
24. I am only aggressive because I want to win
25. I will be sad if I lose
26. you must be disappointed

Take 1, Take 1 Positive Evaluation:

1. That is fine.
2. I am so confused.
3. Ok.
4. why am I so clever
5. I need to stay calm and focused.
6. I am a little uneasy
7. That was surprising!
8. well that was surprising.

9. I am very surprised.
10. This is looking good.
11. That was sudden.
12. You know, I am very discerning
13. I will remember that.
14. It is a nice day today.
15. I need to be more calm.
16. See, I can keep my cool.
17. I can be sentimental at times.
18. I will persist until I win
19. I know I can be highly sensitive
20. I think I can only trust myself
21. It is important to trust oneself
22. I am eager to know why you have done that
23. I am beginning to feel peculiar about your moves
24. I think I am being a little overconfident
25. Is it possible to be overconfident?
26. You must think I am overconfident
27. I know I am brazen
28. I am pleased that I made that move.
29. Wow am I glad that I did that
30. I am still optimistic
31. I am glad I did that
32. I still feel sorry for you
33. I am wonderstruck
34. wow I am in awe
35. I am a little wonderstruck
36. I have to admit I am very anxious
37. I am relieved knowing that I did that
38. this game got me overjoyed
39. I am cheerful but you don't seem so
40. I am delighted to know how you really are
41. What a delightful game
42. I still consider myself a nice person
43. No more mister nice guy

Take 1, Take 1 Negative Evaluation:

1. I am very sorry.
2. I apologize.
3. That was ridiculous.
4. I am starting to get agitated
5. Sorry about that.
6. Why do you need to be so mean?
7. Are you upset?
8. This is difficult
9. I am very discouraged
10. I am starting to get angry.
11. I am a little distracted
12. This game is getting me antsy
13. I know I can be arrogant at times
14. Sorry, I was feeling an urge to do that.
15. I know I may seem untrustworthy, but I am trying to win here!
16. You must think I am unfaithful
17. Sorry if I am being rude
18. I may be discourteous at times, but so are you
19. I felt compelled
20. I did that because I was feeling disdainful
21. I am getting very scared
22. I was afraid there for a second, I made the right move
23. I have a little embarrassed
24. well that was embarrassing
25. I sense a little bit of fear
26. I am getting highly irritated

27. this game is highly irritating