

Continuous Affect Recognition with Different Features and Modeling Approaches in Evaluation-Potency-Activity Space

by

Zhengkun Shang

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, 2017

© Zhengkun Shang 2017

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

Emotions are an essential part of human social interactions. By integrating an automatic affect recognizer into an artificial system, the system can detect humans' emotions and provide personal responses. We aim to build a prompting system that uses a virtual human with emotional interaction capabilities to help persons with a cognitive disability to complete daily activities independently. In this thesis, we work on automatic affect recognition and compare three different types of feature descriptors with support vector machine regression (SVR) and bidirectional long short-term memory (BLSTM) to predict users' emotions in three-dimensional space. We demonstrate the feasibility of further building artificial systems that track users' real-time emotions through BayesACT simulations, a probabilistic and decision-theoretic generalization of Affect Control Theory that learns users' fundamental sentiments during interactions. We would like to understand given virtual humans with distinct emotion characteristics, how and to what extent the user's emotions are affected. In the end, we integrate the affect recognition module into an iterated prisoner's dilemma game, in which a user can play the game against a virtual human. We let a number of participants play the game and test if different facial expressions change the virtual human's strategies during the game.

Acknowledgements

I would like to take this opportunity to thank all the people who made this possible. I would first like to thank my supervisor Prof. Jesse Hoey for his continuous support and guidance. Thanks to Prof. Daniel Vogel and Prof. Mei Nagappan for being my readers and taking the time to read my thesis. Thanks to Jyoti Joshi, with whom I worked on the project and collaborated for a paper. I would also like to thank all my friends and all my CHIL labmates: Areej Alhothali, Josh Jung, Deepak Rishi, and Dan Wang for their support and all the fun we have had together. Additional thanks goes to my boyfriend Jingjie Zheng who always encourage and support me. Finally, I would like to thank my parents, and all my other friends for their constant encouragement.

Table of Contents

List of Tables	vii
List of Figures	viii
1 Introduction	1
2 Related Work	4
2.1 Affective Computing	4
2.2 Affect Recognition	5
2.2.1 Feature Representations	5
2.2.2 Datasets	7
2.2.3 Visual-based Recognition	8
2.2.4 Single Modal and Multimodal	9
2.2.5 Current Problems and Challenges	10
2.3 Affect Control Theory	10
2.4 BayesACT	11
2.5 Applications	11
3 Dimensional Affect Recognition	13
3.1 Semaine Database	13
3.2 Feature Descriptors	14

3.2.1	Action Unit	14
3.2.2	Histogram of Oriented Gradient	17
3.2.3	Felzenszwalb’s HOG	17
3.3	Machine Learning Approaches	18
3.3.1	Support Vector Machine	18
3.3.2	Recurrent Neural Network	19
3.4	Training Process	23
3.5	Result	27
3.5.1	Train and test on clips using SVR and BLSTM	27
3.5.2	Train and test on shuffled images using SVR	28
4	BayesACT Simulations	30
4.1	Background	30
4.2	BayesACT Simulation	32
4.3	Emotion Change in BayesACT Simulation	33
4.4	Sentiment Labels	34
5	Game Integration	36
5.1	Prisoner’s Dilemma Game	36
5.2	System Design	38
5.3	Tests and Discussions	40
6	Conclusion	45
	References	47

List of Tables

3.1	Participants' information and the number of clips extracted from the dataset	15
3.2	Number of clips in training set and test set for each participant	24
3.3	Number of dimensions used for each feature descriptor before and after dimension reduction	27
3.4	EPA prediction with different feature descriptors	28
3.5	EPA prediction on test set with all images shuffled using SVR	29
4.1	Number of simulations for each type of avatar	34
5.1	Payoff matrix in prisoner's dilemma	36
5.2	Payoff matrix in the new iterated prisoner's dilemma game	37

List of Figures

3.1	17 Action Units used in OpenFace	16
3.2	An overview of the HOG process	17
3.3	Kernel types: linear, polynomial and RBF	18
3.4	A typical neural network	20
3.5	An unrolled recurrent neural network	21
3.6	A simple LSTM block	22
3.7	An overview of the training and continuous prediction process	25
3.8	An old version of the training process with all images shuffled	26
3.9	EPA prediction with all images shuffled using SVR (RMSE)	29
4.1	Avatars posterior estimate of the user’s emotions.	33
4.2	Avatars posterior estimate of the user’s transient impressions.	33
4.3	Four word clouds that describe the user’s feeling when talking to different avatars	35
5.1	An overview of the integration system	38
5.2	The game interface of the iterated prisoner’s dilemma game with camera enabled	42
5.3	An overview of the game interface and the facial expressions on the virtual human	43
5.4	Examples of participants playing the game	44

Chapter 1

Introduction

Emotion plays an important role in humans' daily activities. Understanding human emotion is a compelling capability for many types of artificial systems, especially for those that rely on individuals' emotional states. For example, assistive systems help individuals complete simple tasks and they read emotions from persons to provide a more individualized instruction; pain detection systems automatically recognize the pain intensity of patients; tutoring systems provide individualized learning process; and game companions provide a better entertainment experience for users by measuring their engagement.

These studies are in the affective computing field, a cross-discipline of computer science, psychology, and cognitive science. In affective computing, researchers develop intelligent interactive systems that can recognize, interpret, process, and simulate human affects [54]. To recognize emotions, three types of signals can be used: visual signals, audio signals, and biosignals. Visual signals are the cues from users' facial expressions, hand movements, and body postures. Since facial expression is a main way for humans to express their emotions, most studies use facial expressions as visual signals to recognize a person's affective states. Audio signals such as speech rate, pitch range, and vowel duration can also be used for detecting user emotions. Biosignals refer to physiological data such as blood pressure, heart rate, and skin conductance. These different signals provide a way for researchers to detect and analyze user emotions, which is a main step of building automatic affect recognition. Sometimes one of these signals is used (single-modal), and sometimes multiple ones are combined (multimodal).

An automatic affect recognizer aims to detect, process, and analyze human emotions in real-time. There are two streams of emotion recognition: categorical and dimensional facial recognition. The categorical approach classifies human emotion to a limited number

of labeled emotions. Dimensional recognition represents emotions in a multi-dimensional space so more emotions and subtle changes can be detected. In previous studies, researchers focused on basic emotion recognition (e.g. [53]). As the performance of recognizing basic emotions improved and people gradually realized the limitation of categorical labels, more researchers have started working on dimensional recognition, and a coding system to record facial movements called action units [55].

To map facial expressions to multi-dimensional space, we need to extract facial features and use different approaches to build models. Many feature representations exist including low- and high-level features, and different machine learning techniques can be used to train and build models. Since dimensional modeling is a supervised learning, obtaining enough labeled data is crucial. However, rating data is a subjective and laborious task. Multiple annotators can be involved to reduce bias, but combining their ratings to get a higher agreement is still a challenge. Due to the difficulty of data labeling, many databases only contain posed emotion expressions. However, more and more studies suggest that spontaneous facial expressions are different from posed behaviors [40] which moved researchers focus to natural and spontaneous behavior detection. Some other studies also work on coding in action units called Facial Action Coding System (FACS), to capture the complexity of facial expressions objectively.

In dimensional affect recognition, there are two-dimensional and three dimensional approaches. Valence-Arousal is a good example for two-dimensional space. Affect Control Theory (ACT) [46] is a well-known sociological theory which represents behaviors and identities in a three-dimensional space: Evaluation (positive/negative), Potency (powerful/powerless), Activity (active/calm). ACT proposes that people have fundamental sentiments about their identities and they will try to minimize the deflection of transient impressions from fundamental sentiments during interactions. However, the identity is not easy to obtain and sometimes the identities can be changed for people with illnesses [42]. BayesACT is a model [29] based on ACT to solve this problem using a Partially Observe Markov Decision Process (POMDP). It learns interactants' identity and emotions during the interactions between an agent (e.g. a computer) and a client (e.g. a human). The concept is demonstrated by integrating the model into a hand-washing system [33]. In the hand-washing system, BayesACT generates prompts to assist persons with Alzheimer's to complete the hand-washing task. The system computes Power and Activity values based on the expansiveness of the user's hands and the velocity of the user's hand movements, and Evaluation value is set to neutral [33]. For example, if the user moves their hands faster, they will get a high Activity value. When the user expands their hands widely, the Power value will be large. The limitation of the current system is that the EPA measurement of the user's emotions is not accurate enough. A more accurate computation can be

used by obtaining the user’s facial expressions and recognizing their current affect states. Then the system will provide more personalized prompts based on patients’ performance to assist them to complete tasks independently.

The goal of this thesis is to create a basic automatic affect recognizer that can be integrated into various prompting systems so that a virtual assistant can provide more accurate and individualized responses. This thesis will:

1. Compare three different types of feature representations and modeling approaches for dimensional affect recognition.
2. Create an automatic affect recognizer that continuously predicts user emotions in the Evaluation-Power-Activity (EPA) space. (Chapter 3)
3. Demonstrate the feasibility of using BayesACT to simulate interactions between a user and an avatar. (Chapter 4)
4. Integrate the recognizer into a prompting system, in which a user can play the iterated prisoner’s dilemma game with a virtual human. (Chapter 5)

The primary contribution of this thesis is in a demonstration of emotion recognition from the face in EPA space (Chapter 3). The secondary contributions are showing that it can be integrated into BayesACT (Chapter 4), and then into the prisoner’s dilemma game (Chapter 5).

The thesis is structured as follows. Chapter 2 reviews related work regarding affect recognition in the aspects of feature representations, feature extraction and selection, modeling approaches, databases, and applications. It also includes basic concepts of ACT and BayesACT. Chapter 3 introduces the work we do on automatic affect recognition. We compare three different types of feature descriptors with support vector machine (SVM) [11] and bidirectional long-short term memory (BLSTM) [23] to predict affect states in EPA space. More information about these machine learning approaches, feature descriptors, and the Semaine database [38] are introduced in this chapter. Chapter 4 of this thesis demonstrates the feasibility of simulating conversations between a user and an avatar based on BayesACT and determines how and to what extent the user’s emotions are affected. In Chapter 5, we integrate our automatic affect recognizer into an iterated prisoner’s dilemma game, in which the user can play the game with a virtual human. Discussion and future work are in Chapter 6.

Chapter 2

Related Work

Automatic analysis of human affect has attracted an increasing body of research. This chapter reviews the related work regarding the process of affect recognition in the aspects of feature representations, feature extraction and selection, modeling approaches, databases, and applications. It also provides a brief background about Affect Control Theory and BayesACT used in Chapter 4 and Chapter 5.

2.1 Affective Computing

Affective computing is an inter-disciplinary study for developing intelligent interactive systems that can recognize, interpret, process, and simulate human affects or emotions [54]. Emotion is basic and central to human interactions, but most artificial systems are not capable of leveraging this power to better understand human intentions. One of the goals of affective computing is to integrate emotion detection into artificial systems to empower them with the ability of providing more accurate and personalized interaction experience.

Emotional states can be represented using either *categorical labels* or *dimensional models*. *Categorical labels* are a set of discrete human emotions. For example, Ekman [16] used six basic categories to represent human emotions: happiness, sadness, anger, surprise, disgust, and fear. Categorical labels are commonly used for emotion classification, but understandably this approach can only describe a limited number of emotions. In contrast, *dimensional models* use a multidimensional vector to represent an emotion so that it can capture more subtle and complex changes.

Researchers have proposed many dimensional models to represent emotions. Russell [47] first proposed a two-dimensional model, valence and arousal, in which an emotion is described as the extents along two dimensions: pleasant/unpleasant and arousal/sleep. Mehrabian [39], based on the work of Osgood [43], further extended emotions to three dimensions: Pleasure, Arousal, and Dominance (PAD), which is well-accepted in psychological studies. Fontaine et al. [20] also found that two-dimensional space is not sufficient for capturing emotions after they analyzed statistical relations of 144 features like facial expressions with emotion. Affect Control Theory (ACT) [46] uses a three-dimensional vector to represent Evaluation, Power, and Activity. We review more details about ACT in section 2.3.

There have been several competitions gathering researchers to solve open issues on affect recognition. The Audio/Visual Emotion Challenges (AVEC) [51][52][58][57][56] focus on comparing multimedia processing and machine learning approaches for automatic depression detection and dimensional affect analysis. The Facial Expression Recognition (FERA) [60][59][62] evaluates the detection of action unit occurrence and intensity to solve the issue of subject-independent emotion recognition.

2.2 Affect Recognition

An individual's emotional states can be presented in the forms of *visual signals*, *audio signals*, and *biosignals* during human interaction. *Visual signals* often refer to as facial expressions, but body and hand movements can also be cues for visual signals. *Audio signals* include speech rate, pitch range, and vowel duration. *Biosignals* are physiological data such as blood pressure, heart rate, and skin conductance. This thesis focuses mainly on using facial expressions (visual signals) for affect recognition.

2.2.1 Feature Representations

We analyze a person's affect by extracting facial features from videos. A feature representation is encoded information generated from raw data that can be directly used in machine learning tasks. A video can be seen as a frame-by-frame image sequence, and these images are encoded to low or high-level information as feature representations.

Appearance and Shape Representations

Depending on the spatial nature of a feature representation, it can be categorized into either *appearance* or *shape* representations. *Appearance* representations analyzes pixel intensities using textural information; *shape* representations describe shapes explicitly [49].

With respect to *appearance representations*, features can be either low- or high-level:

- *Low-level features* include histograms, edges, and bag of words (features). Histograms such as Local Binary Patterns (LBP) [1] extract local features, represent them in a transformed image, and pool local features of each block with local histograms. Edges are commonly detected by applying a linear Gabor filter on an image with different scales and orientations. Bag of words is originally applied in natural language processing which uses a sparse vector counting the occurrence of words in a text. In computer vision, we count the number of low-level features as “words”. Scale-invariant feature transform (SIFT) [34] is a good example of bag of words that uses a 128-dimensional vector to collect features.
- *High-level features* refer to those that are more interpretable compared to low-level ones. Sparse representation is an example of high-level features, originally used for face recognition. It represents a new signal as a sparse linear combination of the training signals themselves. More recent studies used the same feature in classifying basic emotions and action units [12][37].

Shape representations are less common compared to appearance ones. Researchers use a 2D coordinate to represent landmarks on the face. They can also calculate the distance or the angle between two facial points. The advantage of using shape representations is that the dimensionality is relatively low compared to low-level features and it is crucial for action unit detection.

Convolutional Neural Network

Low-level features are robust against lighting conditions and high-level features can handle identity bias issues [49]. Deep learning such as convolutional neural network (CNN) uses hierarchical representations combining the advantages of low- and high-level features and it uses multiple layers to provide information from low level to high level. CNN chooses a filter as a sliding window to learn the features during training. The first layer of CNN is to detect low-level features such as edges from raw pixels as input and continue learning high-level features such as shape. The pooling layer such as max pooling, is a key aspect

of CNN that applies after getting convolution layers to subsample the previous output. In addition, CNN also has a classifier to do classification after feature extraction.

Dimensionality Reduction

When extracting features, sometimes we obtain a vector with thousands of dimensions, which may exceed the memory capacity in affect recognition. To solve the memory issue, a preprocessing approach is to downsample input images. Moreover, dimensionality reduction can rapidly reduce the number of variables, but still keep the important features. In addition to the pooling method that we have mentioned in CNN, there are two main ways to reduce dimensions: *feature selection* and *feature extraction*:

- *Feature selection* aims to select a subset of features and it is a straightforward way to reduce high dimensional data. It also reduces identity bias since more general vectors are selected during the process. Boosting such as AdaBoost [21] is a widely used technique in feature selection. It is originally designed for prediction on both classification and regression problems, but many researchers use it as a selection technique. Boosting is like a patient consulting multiple doctors and each doctor has a weight based on their reputation. The final decision is determined by the votes from each feature and the weight is based on the feature's effect.
- *Feature extraction* generates new features to represent the original features. This transformation maps the input variables onto a lower dimensional space to reduce dimension, but retain most of the variation in the dataset. The most popular approach is Principal Component Analysis [64], in which the generated variables are the linear combinations of the original variables.

2.2.2 Datasets

The labeled dataset is a crucial part for affect recognition. Recorded clips mainly have two types: posed and natural. Due to the difficulty of data labeling, most of the studies just ask participants to perform basic emotion expressions so that they do not need to manually classify labels. However, more and more studies suggest that the spontaneous facial expressions are different from posed behaviors [40]. For example, it has been shown that temporal and spontaneous smiles have different time duration [10] and brow actions have different action unit intensity and occurrence order [61]. Moreover, since most of the emotion expressions that occur in daily life are not only basic emotions, only classifying

basic emotions is not enough and the datasets about spontaneous or affective behavior are needed.

Having enough labeled data gives a greater accuracy on affect prediction. However, labeling or annotating data is a laborious and subjective task. Data labeling on basic emotions is easy, but it becomes a challenge when we move to dimensional models. By hiring multiple annotators and combining their annotations can reduce annotating errors, but combining them properly to obtain a high agreement is still a difficult task. The common way is to calculate an average value from all the raters. People can also calculate the correlation coefficient among annotators and give each annotator a weight based on their performance. For dimensional annotation, Feel-trace [13] is a widely used tool which allows annotators to move their cursor to rate on two-dimensional space while watching the recordings.

There are many datasets available from different aspects for research uses, and we will only mention a few datasets here. The Cohn-Kanade [36] and the MMI [45] databases are the most widely used databases for facial expression recognition. Both of them contain six basic emotions and action unit annotations. The MMI also has posed and spontaneous facial expressions. The GEMEP [6] collects ten actors portraying 18 emotional states and its subset was used in the FERA challenge. The BU-3DFE database [67] is a 3D facial expression database, which includes 100 subjects with 2500 3D facial expression models.

2.2.3 Visual-based Recognition

Researchers have different opinions about the importance of visual, audio, and physiological signals, but some studies [2] showed facial expressions play the most crucial part in the visual channel and correlate well with the body and voice [68]. Because of the importance of facial expressions in emotion, most researchers use them as the data to work on automatic affect recognition. Current research has two main streams on facial affect recognition: dimensional affect recognition and action unit detection. In previous studies, researchers focused on classifying basic emotions. As the accuracy on classification was high enough and people realized using categorical labels was limited, attention shifted to dimensional affect recognition. For the dimensional models, most of the studies train those dimensions independently. However, since there are correlations between each dimension such as valence and arousal [24], people also considered the relation between dimensions and used different machine analysis like Continuous Conditional Random Fields (CCRF) [4] to train models [49].

Since labeling data is subjective and laborious, action unit uses objective coding stan-

dards to capture the complexity of facial expressions. These high-level features are more interpretable and reliable too. However, trained experts are needed to label images. Therefore, researchers want to use computers to automatically label action units, and they have had some success in detecting both action unit occurrence and intensity. More information about action units will be provided in Chapter 3.

Even though the accuracy of basic emotion classification is high, spontaneous behaviors are also hard to measure with existing techniques. Since an increasing number of datasets contain natural and spontaneous behaviors instead of posed facial expressions, researchers are trying to analyze spontaneous behaviors using different machine learning approaches.

Machine Learning Approaches

Selecting a proper machine learning approach is crucial for automatic affect recognition. Support Vector Machine (SVM) [11] is a widely used technique for classification and Support Vector Regression (SVR) is for regression problems. Structured SVM is a variant of SVM that is used for interdependent and structured output space. The hidden Markov Model (HMM) is a good example of modeling the temporal variation of facial expressions with SVM or Boosting to improve prediction.

Researchers have also applied CNN for feature extraction and classification. Since the recurrent Neural Network (RNN) can handle sequential information, it is a good fit for video processing. With the special structure of RNN, long-short term memory (LSTM) [28] is widely used in natural language processing and computer vision. It is also compared with SVM on both classification and regression [41]. Since we also use LSTM in our work, we will provide more detailed discussions about RNN and LSTN in Chapter 3.

2.2.4 Single Modal and Multimodal

Most studies treat different signals separately, but an increasing number of studies such as [5] suggested that grouping visual, audio, and biosignals together can reduce prediction errors. The recent AVEC competition [56] also provides both visual and audio data and encourages participants to use multiple modalities to build affect recognizers.

However, how to combine different channels is a problem. There are two common ways to combine different data: feature-level and decision level. Feature-level combines different features together and then puts them into a training set. However, different modalities provide different formats of data. Grouping them together may also influence

the performance since the dimension of features also increases. In the decision-level, every modality is treated independently, and the results are combined in the end. Since different modalities actually are correlated to each other but have different influence, a correlation needs to be added. Even though many modalities show a better result than a single-modal, how and when to combine different modalities gives a direction of future work.

2.2.5 Current Problems and Challenges

Although many researchers show that they have already achieved a high accuracy on affect recognition, there are still many problems and challenges. There is a trend from studying posed expressions to spontaneous behaviors and from using single-modal to multimodal. However, a few conditions can still have a negative effect on affect recognition, such as light conditions, head-pose variations, and annotation errors.

The current studies always assume the labeled data is correct. However, for dimensional affect recognition, labeling data is subjective depending on the annotator’s opinion on the emotion. Averaging the values among multiple annotators can reduce bias, but finding a more proper way to combine those annotations is still a challenge. For visual signals, a main issue is person-independent prediction – the training works well on the same participants in the training set, but sometimes the model does not fit on a new person. At the same time, the trained model is also sensitive to light conditions and head-pose variations.

2.3 Affect Control Theory

A sociology theory called Affect Control Theory (ACT) uses a three-dimensional vector to describe emotions. The basis vectors of the affective space are called Evaluation (pleasant/unpleasant), Potency (powerful/powerless), and Activity (exciting/calm) (EPA).

ACT represents social behaviors with an Actor-Behavior-Object model in which an actor behaves towards an object. Each of these elements has an EPA score, ranged from -4.3 to 4.3 . EPA profiles can be measured with a survey where annotators rate identities and behaviors on numerical scales [44]. People with similar cultures reach a high agreement about EPA values of identities and behaviors, and averaged EPA ratings are extremely stable over time even with a few dozen participants [27]. For example, the EPA value of a student is $[1.49, 0.31, 0.75]$ describes students as good, not really powerful, and a bit active. A teacher is seen as $[2.45, 1.75, 0.29]$, which is considered a bit better, powerful, and less active compared with a student.

ACT proposes that the interactions are always adhered to by a psychological need of minimizing the deflection between the fundamental sentiment and the transient impression. Fundamental sentiments are representations of social objects, such as interactants identities and behaviors or environmental settings in EPA space. Transient impression, which is also a three-dimensional vector in EPA space, results from the interaction in a social event which may cause deviation in the identity or behavior of the interactants from their corresponding fundamental sentiment. The Euclidean distance between the fundamental sentiment and the transient impression is the deflection. A small deflection will let the interactions run smoothly. For example, when a mother plays with a child, the behavior is described as $[3.4, 1.8, 0.9]$. Since a mother has an EPA value of $[2.9, 1.5, 0.6]$, the small deflection makes both of them comfortable. However, if a mother beats a child, in which the action has $[-1.0, 3.5, 2.2]$, the behavior has a large deflection from the EPA of a mother, and the large deflection is humans seek to avoid based on the ACT principle.

2.4 BayesACT

BayesACT [29] is a probabilistic and decision-theoretic generalization of ACT. It keeps multiple hypotheses about both identities and behaviors as a probability distribution, and uses an explicit utility function to make value-directed action choices. This allows the new model to generate affectively intelligent interactions with people by learning about their identity, predicting their behaviors according to the ACT principle, and taking actions that are simultaneously goal-directed and affect-sensitive. Moreover, BayesACT allows ACT to model more complex affective sentiments, including ones that are multimodal. It has been used in the COACH system [30], a POMDP-based agent that can assist persons with dementia by monitoring the person and providing audio-visual cues when the person needs help. The concept is demonstrated by integrating the model into a hand-washing system [33]. This combination creates an updated COACH system that can choose an appropriate action with an EPA output that minimizes the deflection based on the ACT principle.

2.5 Applications

Integrating affect recognizers into artificial systems are demonstrated to offer more enriched interaction experience in various applications:

- In clinical studies, the intensity of pain is reported by a patient or an observer, but is often biased by its subjectivity. It is also limited due to the effort needed so that the pain could not be monitored continuously over time. Lucey et al. [35] used facial action units to automatically detect pain in videos.
- Emotions are also highly relevant to driving. For example, Healey and Picard [25] used physiological signals to detect stress levels during driving tasks; Eyben et al. [18] used LSTM to build an online driver distraction detector.
- Emotions can also be used for providing better gaming experience. Sanghvi et al. [48] proposed iCat robot that acts as a game companion to play chess with children, detecting postures and movements to analyze their engagement. Savva and Bianchi-Berthouze [50] built a system to recognize emotional states from body movements while participants playing a tennis video game.
- Tutoring systems can offer an appropriate strategies based on learners' facial expressions. For example, AutoTutor [15] is an intelligent tutoring system that can process users' emotional states and select tutor actions that maximize learning effect.

Chapter 3

Dimensional Affect Recognition

To enable emotional interaction between users and prompting systems, we first need to predict users' emotions in real-time. We use Evaluation, Potency, and Activity (EPA) to quantify users' emotions. In this chapter, we train EPA models using two machine learning approaches: the support vector machine and the recurrent neural network to predict users' emotions from their facial expressions. Three feature descriptors are extracted frame by frame from videos of the Semaine database. We then compare the performance of these feature descriptors.

3.1 Semaine Database

The Semaine database [38] is used in our studies as the source of data. This database provides extensive annotated audio and visual recordings of a person interacting with an emotionally limited agent, or sensitive artificial listener (SAL), to study natural social behavior in human interaction. The Semaine database provides three SAL scenarios:

- Solid SALs are human operators playing the roles of the characters directly
- Semi-automatic SALs are systems, controlled by an unseen human operator, interacting with a user
- Automatic SALs are systems selecting sentences and words automatically

We extract all the annotated videos from the database, each of which is a conversation between a user and an agent. The agent is asked to act in one of the four solid SAL avatars in each video: Poppy is happy and tries to make the user happy; Spike acts angry; Obadiah is sad and depressed; and Prudence is sensible and even-tempered. The video recordings are transcribed and annotated frame by frame by 6 to 8 raters into five affective dimensions: Valence, Power, Activation, Anticipation, and Intensity. The first three dimensions constitute our EPA space in the studies. Since only a few videos of the agent are annotated and the agent is acting in different characters, we only use user’s clips instead, which gives us 93 clips with 20 persons involved. The video is recorded at 49.979 frames per second and all of emotions are annotated from -1 to 1.

The specific information of the collected clips is shown in Table 3.1. The data has a 60% female and a 40% male population in total, interacting with the agent (avatar). Each video clip records the conversation between one participant and one agent enacting one type of avatar. Most of the participants talk to one agent enacting four different avatars. Therefore, there are four video clips for each participant, except for two participants (ID: 7, 16) talking to multiple agents. The data for participant 4, 9, 16, and 19 are not complete. In Table 3.1, we report the 20 participants’ ID, gender, number of clips they interact with avatars, and type of avatars they talk to.

3.2 Feature Descriptors

We first experiment with three commonly used visual feature descriptors extracted from raw face images and compare their accuracy in predicting a person’s Evaluation, Potency, and Activity (EPA) scores. These descriptors are the Action Unit [55], the Histogram of Oriented Gradient [14], and Felzenszwalb’s HOG [19]. We select these descriptors because they are commonly used in computer vision and they characterize human faces from different aspects.

3.2.1 Action Unit

The Action Unit (AU) is a high-level feature descriptor that detects facial movements. The corresponding coding system, called the Facial Action Coding System (FACS), was developed by Ekman and Friesen [17] to encode movements of facial muscles. FACS defines 44 AUs, 30 AUs of which are related to specific facial muscles. FACS is a standard way to describe facial expressions, since it is possible to code nearly any facial expression by

Table 3.1: Participants' information and the number of clips extracted from the dataset

Participant ID	Gender	# Clips	Happy	Angry	Sensible	Sad	Video ID
2	M	4	1	1	1	1	46-49
3	M	8	2	2	2	2	19-22, 40-43
4	M	3	1	1	1	0	25-27
5	F	4	1	1	1	1	13-16
7	F	8	2	2	2	2	34-37, 76-79
8	F	4	1	1	1	1	52-55
9	F	2	1	0	1	0	58,60
10	M	4	1	1	1	1	70-73
11	M	4	1	1	1	1	64-67
12	F	4	1	1	1	1	82-85
13	M	4	1	1	1	1	88-91
14	F	4	1	1	1	1	94-97
15	F	4	1	1	1	1	100-103
16	F	11	3	2	3	3	2-5, 8-11, 29-31
17	M	4	1	1	1	1	106-109
18	F	4	1	1	1	1	112-115
19	M	5	2	1	1	1	118-122
20	F	4	1	1	1	1	125-128
21	F	4	1	1	1	1	131-134
22	F	4	1	1	1	1	137-140

deconstructing a facial appearance into multiple AUs. Moreover, AUs can be annotated by occurrence and intensity. Occurrence is a binary number to show if an AU is visible in the face and intensity gives a real number to describe how intense an AU appears. Since AUs can appear either singly or in combination, they can be used to classify emotions. For example, the Emotional Facial Action Coding System (EFACS) [22] considers only emotion-related facial expressions. If both AU 6 (cheek raiser) and AU 12 (lip corner puller) appear on a user’s face, we can predict the user is happy.


















AU1  Inner Brow Raiser	AU2  Outer Brow Raiser	AU4  Brow Lowerer	AU5  Upper Lid Raiser	AU6  Cheek Raiser
AU7  Lid Tightener	AU9  Nose Wrinkler	AU10  Upper Lip Raiser	AU12  Lip Corner Puller	AU14  Dimpler
AU15  Lip Corner Depressor	AU17  Chin Raiser	AU20  Lip Stretcher	AU23  Lip Tightener	AU25  Lips Part
AU26  Jaw Drop	AU45  Blink			

Figure 3.1: 17 Action Units used in OpenFace

In our studies, we leverage OpenFace [3], an open-source framework, to localize face positions and extract AUs from aligned facial images. OpenFace is able to recognize 17 AUs and the description of each AU is shown in Figure 3.1. We use the intensity of AUs to obtain more facial information and the range is from 0 to 5.

3.2.2 Histogram of Oriented Gradient

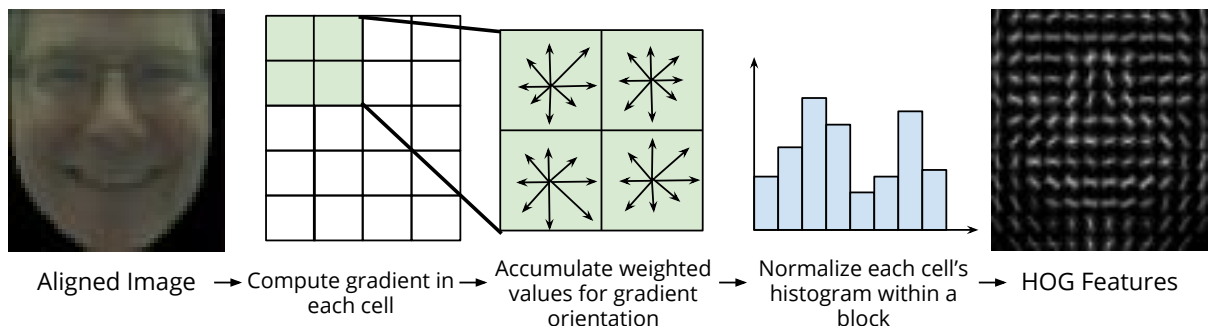


Figure 3.2: An overview of the HOG process

The Histogram of Oriented Gradient (HOG) is a typical low-level feature descriptor proposed by Dalal and Triggs [14] for human detection in 2005. HOG splits an image into a number of non-overlapping cells. For each cell, it computes histograms of gradients and discretizes them into nine orientation bins. After accumulating weighted values, HOG normalizes each cell's histogram with the total energy of the four 2×2 blocks containing this cell. Then a 36-dimensional feature vector is generated from this image. Those parameters including cell size, number of orientation bins, and block size, can be chosen empirically. This feature descriptor is commonly used for localizing face positions paired with an SVM classifier. Figure 3.2 shows an overview of the HOG process.

We use HOG descriptor from the OpenCV library [31] to extract HOG features from aligned facial images. Initially, we use default parameters with cell size 8×8 and block size 16×16 , but soon we find the high dimensional descriptor is a problem when we reduce dimensions and train models later. Therefore, we change the cell size to 16×16 and the block size to 64×64 to obtain 5184 dimensions to balance the number of dimensions and the amount of information from input images.

3.2.3 Felzenszwalb's HOG

Felzenszwalb's HOG (FHOG) is a variant of HOG proposed by Felzenszwalb et al [19]. It has been shown to achieve superior performance to the original HOG features. Compared to HOG, FHOG reduces the HOG feature space using principal component analysis. This makes it possible to use fewer parameters in its models to speed up detections and learning

processes. Overall, they derived a 13-dimensional alternative feature to replace the original 36-dimensional feature used with the same parameters shown above, but achieved the same performance as the original result. In our work, we use the dlib library [32] to extract FHOG features and the dimension of the feature descriptor is less than one thousand, using the same cell size. Thus, we decrease the cell size to 8×8 for more information, which gives us a 4464 dimensional descriptor instead.

3.3 Machine Learning Approaches

After extracting features from videos, we apply the machine learning models using two different approaches to predict EPA values: the Support Vector Machine (SVM) and the Recurrent Neural Network (RNN). Those approaches are selected because they are commonly employed in work reporting on continuous affect prediction. The results of comparing different types of features will be more convincing by implementing different training approaches.

3.3.1 Support Vector Machine

The Support Vector Machine (SVM) [11] is a common machine learning approach for classification and regression analysis in emotion prediction. SVM uses labeled training data to generate an optimal hyperplane that gives the minimum distance to datasets. Since dimensional affect prediction is a regression problem, we use SVM-Regression (SVR) in our work.

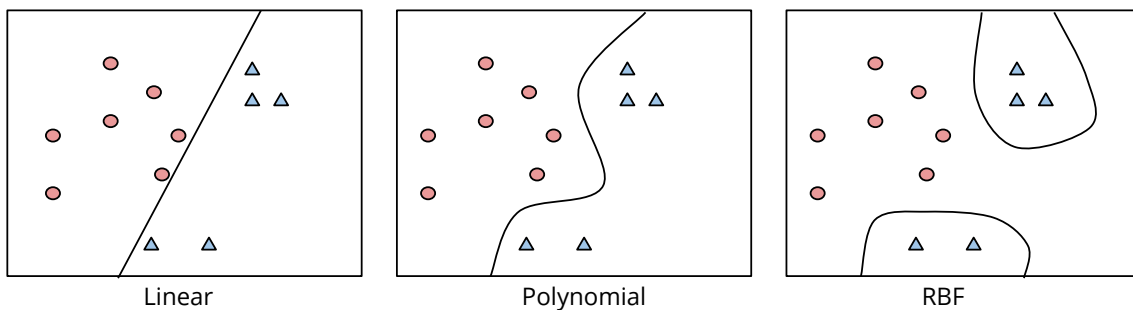


Figure 3.3: Kernel types: linear, polynomial and RBF

There are four parameters in SVR: kernel type, C, epsilon, and gamma:

- The kernel type can be linear, polynomial, or radial basis function (RBF), which represent different ways of interpreting the relations in datasets. A kernel is a function that computes the similarity between two inputs and provides the result to a machine learning algorithm. Figure 3.3 shows these three kernel types. A linear kernel type is a linear function used to find a linear separation of the data. Since most of the data are more complex and not linearly separable, a more complicated model is needed here. RBF achieves a better result than other kernel types and it is widely used in SVM.
- The parameter C trades off misclassification of training data against the simplicity of the model surface. A large C aims to find more support vectors to predict every input correctly, but the model surface will be complex and the variance will be high as well.
- The gamma means how far the influence of a training data reaches. The model is sensitive to the gamma. If the gamma is small, the model will be constrained and cannot describe the model's shape correctly.
- The epsilon in the error function specifies a bound to ignore errors in this range. There is no penalty in the training loss function if the predicted value is within a distance from the actual value. Otherwise, anything beyond the bound will be penalized. If the epsilon is large, more errors will be ignored and the model will be less accurate. However, a small epsilon will end up with a large number of support vectors.

3.3.2 Recurrent Neural Network

In addition to SVR, we also use the recurrent neural network (RNN) to compare different types of features. To define a neural network, Maureen Caudill in his work *Neural Networks Primer* [7] gives a definition of a neural network as “a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs”. A neural network is organized in layers, and typically it has an input layer, a number of hidden layers, and an output layer. Layers are grouped by a number of interconnected nodes and nodes contain activation functions. A typical neural network is shown in Figure 3.4, which x_i is the input value, o_l is the output, y_j and z_k are the hidden variables, and w shows the weight for each connection.

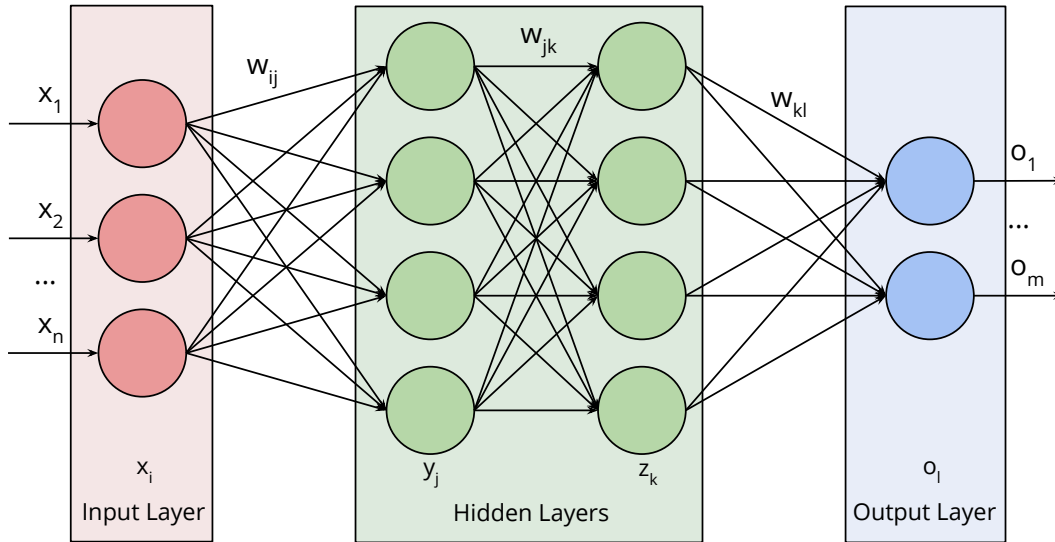


Figure 3.4: A typical neural network. The variable x_i is the input value, o_l is the output, y_j and z_k are the hidden variables, and w shows the weight for each connection.

Compared to the traditional neural network where all inputs are independent of each other, RNN addresses this issue that it predicts every next value based on the previous information. In fact, RNN uses loops to capture information into its memory and passes previous knowledge to predict the next value. In Figure 3.5, x is the input, o is the output at each step, s denotes the hidden state, and w shows the weight. Figure 3.5a shows the traditional RNN and it can be unrolled into a chain as shown in Figure 3.5b. Based on its special structure, RNN has been applied to many areas: image captioning, speech recognition, translation, image processing, etc. However, RNN is not capable of learning long-term dependencies. Sometimes only recent information is needed to perform the prediction. As the length of sequence grows, it will be more difficult for RNN to learn to connect the information. To solve this problem, Hochreiter and Schmidhuber [28] introduced LSTM which is a special kind of RNN, and now it is widely used in a variety of problems.

Figure 3.6 shows the structure and function of LSTM. The variable x_t denotes the input vector at t time stamp and h_t is the output of current block. LSTM uses a cell state c_t to denote the memory from current block and makes three gates: input, forget, and output to control the cell state. There are two operations: ‘+’ denotes the concatenation and ‘*’ is the element-wise multiplication.

The first step of LSTM is to decide which previous information needs to be erased by

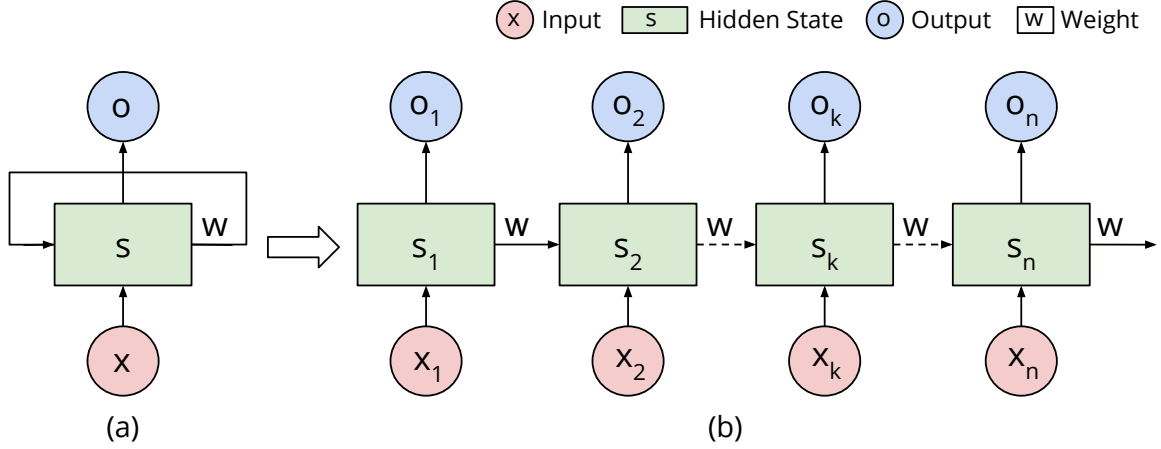


Figure 3.5: (a) is a traditional RNN. (b) is this RNN unrolled into a chain.

the forget gate. A gate is a filter to add or remove information to the cell state. It is a sigmoid layer and the output is between zero and one. If the output value is zero, it means no information passes, while a value of one means the gate keeps all the information. The formula to update forget gate f at t time stamp is shown below, which W_{x_f} , W_{h_f} , and W_{c_f} are the weights for input x , output h , and cell state c . The sigmoid function is represented as σ .

$$f_t = \sigma(W_{x_f}x_t + W_{h_f}h_{t-1} + W_{c_f}c_{t-1} + b_i) \quad (3.1)$$

The second step is to choose which new information is added to the cell. LSTM combines two values together to update the cell. One value uses the input gate layer to decide how much scale we need to update the state, and the other creates a tanh layer to generate a new candidate value to be added to the cell state. The cell state is put through a tanh layer so that only a final value between -1 to 1 will be generated.

$$i_t = \sigma(W_{x_i}x_t + W_{h_i}h_{t-1} + W_{c_i}c_{t-1} + b_i) \quad (3.2)$$

$$c_{temp} = \tanh(W_{x_c}x_t + W_{h_c}h_{t-1} + b_c) \quad (3.3)$$

To calculate a new cell state, we multiply the previous state by the forget gate to remove the memory that we decide to forget, and then add the candidate value, scaled by how much we choose to update.

$$c_t = f_t * c_{t-1} + i_t * c_{temp} \quad (3.4)$$

Simultaneously, the output gate layer chooses which part of cell state to output using a sigmoid gate. Then we run a tanh layer on the cell state to push the range to be $[-1, 1]$,

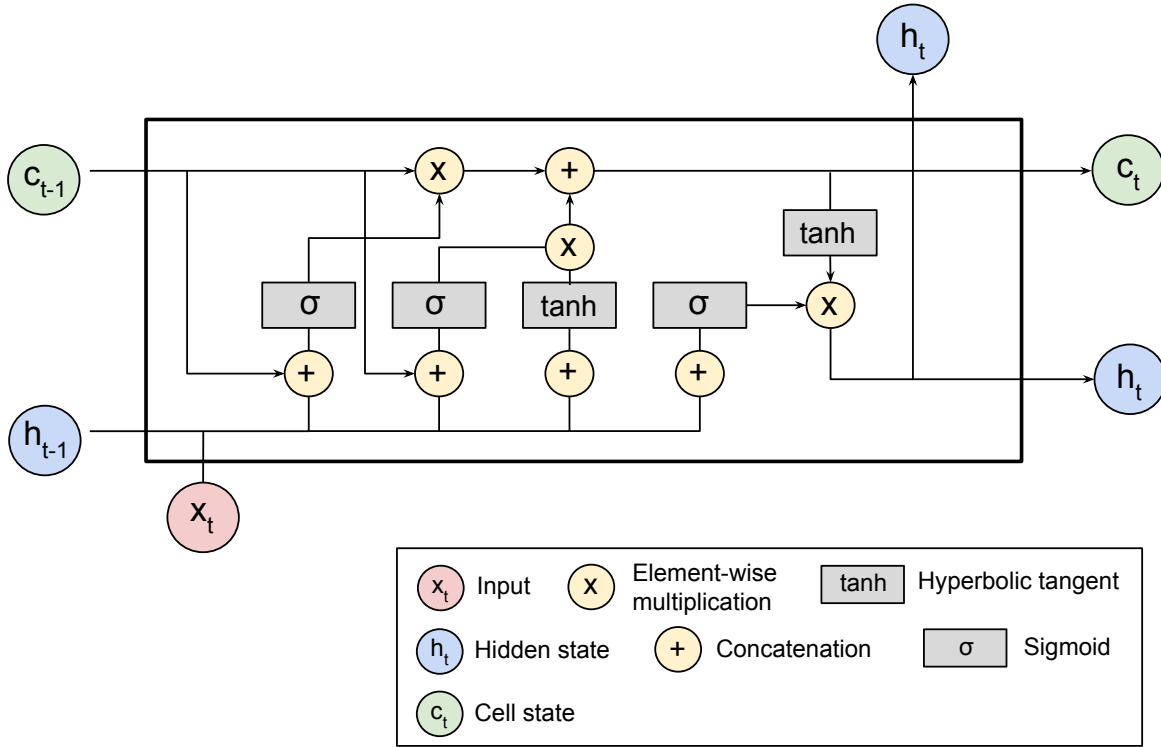


Figure 3.6: A simple LSTM block

and multiply it by the output of the sigmoid layer.

$$o_t = \tanh(W_{x_o}x_t + W_{h_o}h_{t-1} + b_o) \quad (3.5)$$

$$h_t = o_t * \tanh(c_t) \quad (3.6)$$

Bidirectional long short-term memory (BLSTM) is a special structure of LSTM in which two hidden layers of two opposite directions are connected to the same output. As a result, the output layer in BLSTM obtains both past and future information.

There are many parameters in BLSTM, and the main parameters are:

- The number of hidden units represents the learning capacity of a neural network. If the number of hidden units is small, it will not capture a complex model well, and it will lead to a high error rate. However, a large number of hidden units takes longer to train and it also runs the risk of over-fitting.

- The learning rate is the step size used to find the optimized value. If the learning rate is large, the cost function converges quickly, but it will probably miss the best model and never reach the minimum because of the large step size. When a small learning rate is applied, the function takes a longer time to converge and it needs more iterations to reach the minimum.
- The optimizer is an optimization algorithm used in BLSTM to optimize the loss function. Many optimizers can be used in LSTM such as Stochastic Gradient Descent, Adagrad, RMSProp, Adadelta, and Adam, and each of them has a different way to treat the learning rate.
- The dropout rate is a technique in which a neural network randomly chooses some neurons to ignore. Therefore, the information stored in the selected neurons are removed and will not contribute any value during the training. The effect of the dropout is that a trained model will be less sensitive to specific neurons, and it will generalize better to avoid over-fitting.
- The batch size and the epoch size are also important during the training. The sequences of training data are grouped into several batches and each batch is trained at one time. The epoch is the number of iterations in the training process. A large epoch is needed if the function takes longer to converge and reach the minimum, but if the iteration is significantly large, it may miss the minimum point and the result will worsen.

3.4 Training Process

For each feature descriptor, we employ the same workflow illustrated in Figure 3.7 for training and testing models. At first, we randomly split 93 video clips into a training set and a test set. The training set has 75 clips and the test set has 18 clips. The number of clips for each participant in the training set and in the test set is listed in Table 3.2. For each image in a video, we localize and align the face in the image, then extract the feature using one of the three descriptors. Feature descriptors such as HOG and FHOG can have thousands of variables in their feature space, therefore we reduce their feature dimensions with dimension reduction. After dimension reduction, we apply the data in the training set for tuning and in the test set for evaluating the model using SVR and BLSTM.

We initially employ all 93 annotated videos from the Semaine database for model training and this translates to well over one million images. However, we soon find this

Table 3.2: Number of clips in training set and test set for each participant

Participant ID	Number of clips in the training set	Number of clips in the test set	Total
2	4	0	4
3	6	2	8
4	3	0	3
5	3	1	4
7	5	3	8
8	4	0	4
9	2	0	2
10	2	2	4
11	4	0	4
12	3	1	4
13	4	0	4
14	4	0	4
15	4	0	4
16	9	2	11
17	3	1	4
18	3	1	4
19	3	1	4
20	2	2	4
21	3	1	4
22	3	1	4

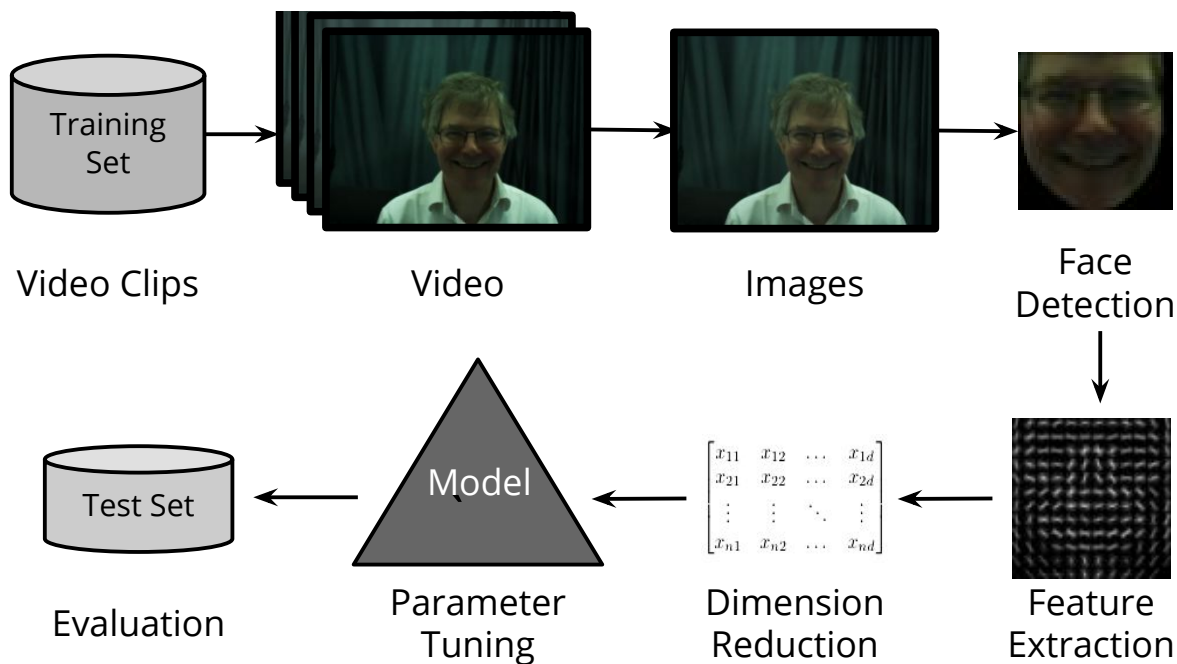


Figure 3.7: An overview of the training and continuous prediction process

large amount of data exceeds the memory capacity during the training of the models. We therefore sample one frame every 0.4 seconds from the videos, resulting in 62,916 images in total. To further optimize the time required for training, we use principal component analysis (PCA) to reduce the feature space. The first k eigenvectors that contain at least 80% variance are selected. Table 3.3 shows the dimension number after feature extraction and after applying PCA.

For each feature descriptor, three models that correspond to evaluation, potency, and activity are trained. We use SVR and BLSTM for training all models. At first, we shuffle all images before training to reduce bias to a single video for SVR. Since BLSTM takes sequences of data to train models, we cannot shuffle the images in each video in this case. Therefore, we choose to randomly pick clips instead. In this case, the training set has 75 clips and the test set has 18 clips. However, we also divide the data set into a training set with 50,601 images (80%) and a test set with 12,315 images (20%) for another experiment of SVR as shown in Figure 3.8.

We use LibSVM [8] as a library to implement SVR. The kernel type is RBF. Other parameters: C , γ , and ϵ in loss-function, are optimized using grid search with five-fold cross

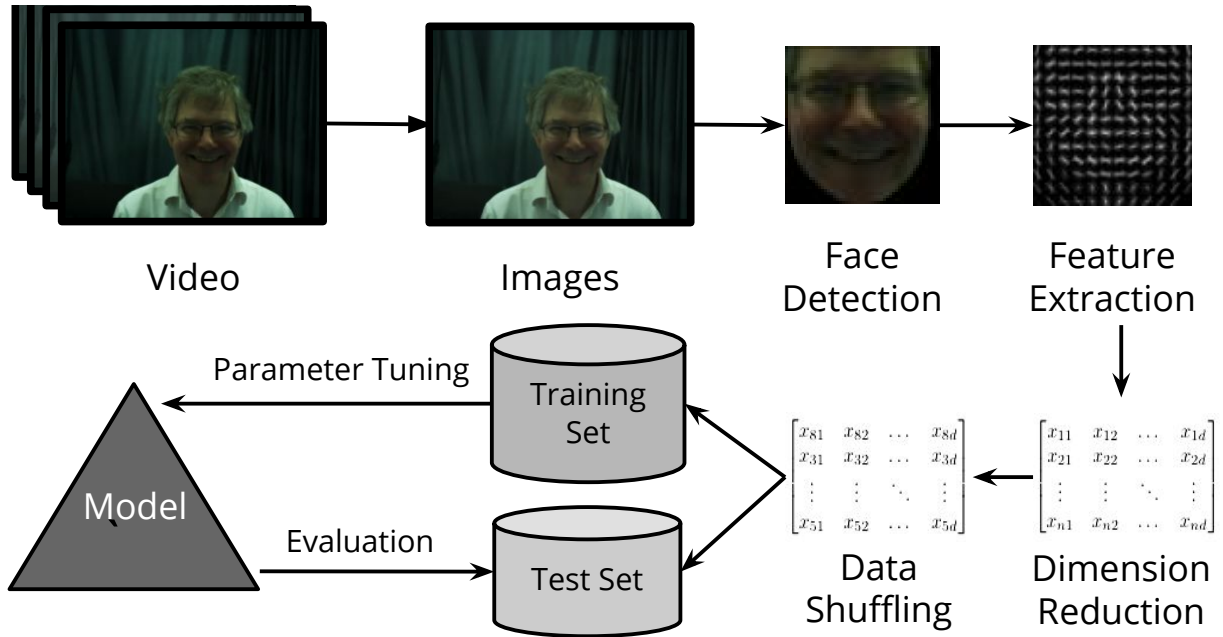


Figure 3.8: An old version of the training process with all images shuffled

validation. For example, the complexity of the SVR is optimized by the best performance in the five-fold cross validation among [0.001, 0.01, 0.1, 1, 10, 100]. The five-fold cross validation is to split the training set to 5 sets and each contains 15 clips. Each time we use four different sets (60 clips) to train the model and test on the left 15 clips. We continue the same step for five rounds until all the five sets have been used to test performance. To evaluate a performance for different parameters, we use the root-mean-square error (RMSE) as our metric.

$$\sqrt{\frac{\sum_{k=1}^n (y_k - y'_k)^2}{n}} \quad (3.7)$$

The ground truth for the k th value is y_k , y'_k is the predicted value, and n is the total number of data we have. We choose the best parameter as the least RMSE from cross validation. The other parameters γ and ϵ are also chosen similarly.

To implement BLSTM, we use Keras [9], a deep learning library for Tensorflow to train models. We set BLSTM with one layer as our neural network structure and RMSProp is used as the optimization function. The parameters used are: the number of hidden units, learning rate, dropout rate, and batch size tuned using grid search. To improve the efficiency of grid search, we only change one parameter to a range of numbers at each

Table 3.3: Number of dimensions used for each feature descriptor before and after dimension reduction

Feature Descriptor	Before PCA	After PCA
AU	17	7
HOG	5184	49
FHOG	4464	153

time and check the performance for each parameter. Then during grid search, we select a subset in this range to run fewer combinations of different parameters with five-fold cross validation. For example, we set the number of hidden units among [32, 64, 128, 256, 512]. After identifying how it works, we set this parameter to [64, 128, 256] during grid search. Moreover, we set the number of iterations to 50, but an early stopping is used in case the function passes the best model before 50 iterations. In Keras, all the sequences should have the same length for training, so we also add padding zeros and mask to the end of each video. The mask is used here to ignore those paddings. After grid search, the parameters with the best performance in five-fold cross validation are used to evaluate the test set.

3.5 Result

To evaluate a model’s prediction performance, we choose the parameters with the least RMSE from cross validation to train the models. The fitted models are evaluated with the test set. We also calculate the Pearson Correlation Coefficient (PCC) as a reference. RMSE evaluates how much predicted values and the ground truth vary; PCC calculates the covariance of the predicted values and the ground truth values.

3.5.1 Train and test on clips using SVR and BLSTM

Table 3.4 shows the prediction performance for applying SVR or BLSTM with the following three feature descriptors: AU, HOG, and FHOG after applying PCA. Note that the AU descriptor has only 17 dimensions, applying PCA is not necessary, but we still provide AU with PCA as a condition for ease of comparison. This, as expected, results in a worse prediction performance due to 20% loss of variance. To summarize the result:

Table 3.4: EPA prediction with different feature descriptors

Dimension	Feature Descriptor	SVR				BLSTM			
		Training Set		Test Set		Training Set		Test Set	
		RMSE	COR	RMSE	COR	RMSE	COR	RMSE	COR
Evaluation	AU	0.251	0.127	0.272	0.235	0.236	0.351	0.258	0.278
	AU+PCA	0.252	0.081	0.273	0.205	0.243	0.290	0.259	0.380
	HOG+PCA	0.251	0.082	0.274	0.213	0.230	0.453	0.246	0.465
	FHOG+PCA	0.251	0.161	0.277	0.121	0.235	0.472	0.251	0.444
Potency	AU	0.225	0.176	0.233	0.189	0.203	0.450	0.213	0.381
	AU+PCA	0.227	0.024	0.234	0.081	0.215	0.335	0.224	0.304
	HOG+PCA	0.228	0.040	0.235	0.122	0.223	0.371	0.218	0.401
	FHOG+PCA	0.229	0.087	0.233	0.196	0.222	0.303	0.217	0.429
Activity	AU	0.220	0.303	0.221	0.352	0.213	0.371	0.207	0.431
	AU+PCA	0.231	0.181	0.229	0.339	0.199	0.522	0.201	0.517
	HOG+PCA	0.219	0.336	0.216	0.414	0.188	0.601	0.205	0.471
	FHOG+PCA	0.217	0.350	0.214	0.444	0.192	0.550	0.201	0.493
(Average)	AU	0.232	0.202	0.242	0.259	0.217	0.391	0.226	0.363
	AU+PCA	0.237	0.095	0.245	0.208	0.219	0.382	0.228	0.400
	HOG+PCA	0.233	0.153	0.242	0.250	0.214	0.475	0.223	0.446
	FHOG+PCA	0.232	0.199	0.241	0.254	0.216	0.442	0.222	0.455

- There is not much difference among all the three features' results with three significant digits. However, to compare the three feature descriptors, FHOG has the best averaged accuracy of EPA. HOG is slightly better than AU on averaged RMSE in EPA. A probable reason for this is that the AU recognition can be noisy as well.
- Between SVR and BLSTM, we can see BLSTM obtains a better evaluation than SVR, among all the conditions.
- Among the three EPA dimensions, Activity is shown to have the best accuracy on the test set. This is probably because Activity is easier to detect from facial expressions than Evaluation and Potency.

3.5.2 Train and test on shuffled images using SVR

Table 3.5 and Figure 3.9 show the result when SVR is applied and all the images are shuffled. The same four conditions are still applied here. The result shows a large improvement compared with the evaluations above. Since all the images are shuffled before training, the images in the same video appear both in the training set and the test set, which increases

Table 3.5: EPA prediction on test set with all images shuffled using SVR

Feature Descriptor	Evaluation		Potency		Activity	
	RMSE	COR	RMSE	COR	RMSE	COR
AU	0.210	0.575	0.185	0.591	0.179	0.631
AU+PCA	0.242	0.322	0.215	0.424	0.209	0.424
HOG+PCA	0.146	0.822	0.142	0.785	0.120	0.854
FHOG+PCA	0.127	0.868	0.128	0.885	0.107	0.829

the performance. The poor prediction accuracy of the unshuffled models is not because the images of the same participant appear only in the test set. We list the number of clips in the training set and in the test set for each participant as shown in Table 3.2, and all the participants appear in the training set. The result also provides a stronger proof of the previous conclusion. It proves that FHOG obtains much better evaluations than other three conditions. AU with PCA decreases the performance and HOG gives a better result than AU. Moreover, Activity is still the easiest dimension to predict among all the dimensions.

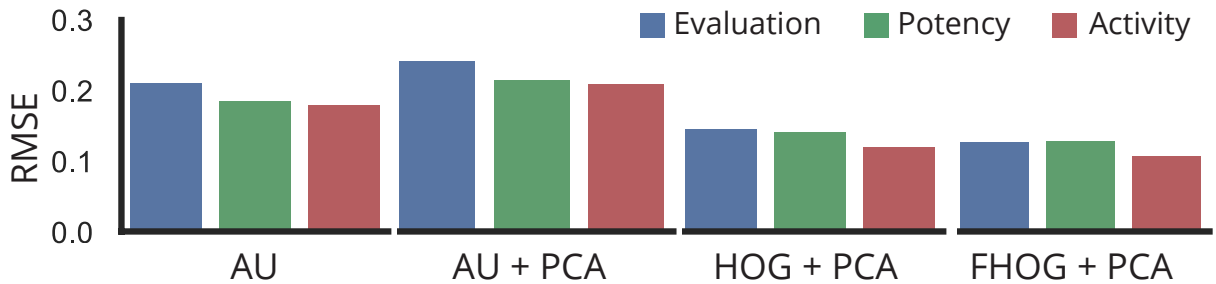


Figure 3.9: EPA prediction with all images shuffled using SVR (RMSE)

Chapter 4

BayesACT Simulations

With facial features transformed into the EPA space that characterizes the users' sentiments, we demonstrate the feasibility of further building artificial systems that track users' real-time emotions through BayesACT simulations. The goal of this chapter is to use BayesACT model to simulate conversations between a user and an avatar, and see whether the avatar can perceive the affective information of users.

4.1 Background

Chapter 2 of this thesis introduces basic information about ACT and BayesACT. In this section, we will provide more background on BayesACT. ACT has a prerequisite that the two interactants identities are known to each other. However, this prerequisite is a challenge to achieve when users' identities are unknown or dynamic. BayesACT [29] is the model to tackle this problem that it keeps multiple hypotheses about both identities and behaviors as a probability distribution using Bayesian reasoning, and uses an explicit utility function to make value-directed action choices.

In this section, we use capital symbols (F , T) to denote variables, small symbols (f , τ) to denote the values of these variables, boldface symbols (\mathbf{F} , \mathbf{T}) to denote a set of variables, and primes (X' , x') to denote the variable or the value of the variable after a time step. $\mathbf{F} = \{F_a, F_b, F_c\}$ is a nine-dimensional vector that denotes a set of fundamental sentiments for actor (agent), behavior, and object (client). Each fundamental sentiment is a vector of EPA which represents social objects such as identities and behaviors in EPA space. For example, if a tutor compromises with a student, \mathbf{f} will be (1.506, 1.445, -0.18,

1.884, 1.208, -0.152, 1.487, 0.304, 0.759). The first three numbers denote the identity of the tutor after this event; the middle three values are EPA of the action “compromise with”; and last three are the identity of the student. Similarly, a set of transient impressions is denoted as $\mathbf{T} = \{T_a, T_b, T_c\}$. Therefore, the deflection D is the sum of squared Euclidean distance between fundamental sentiments \mathbf{F} and transient impressions \mathbf{T}

$$D = \sum_{i=1}^9 w_i (\mathbf{f}_i - \boldsymbol{\tau}_i)^2 \quad (4.1)$$

where w_i is the weight. A small deflection will let the interactions run smoothly. For example, when a mother plays with a child, the behavior “play” is described as [3.4, 1.8, 0.9] and a mother has an EPA value of [2.9, 1.5, 0.6]. The small deflection makes both of them comfortable. However, if a mother beats a child, in which the action “beat” has [-1.0, 3.5, 2.2], the behavior has a large deflection from the EPA of a mother, which is humans seek to minimize the deflection based on the ACT principle. Therefore, both the mother and the child may have negative emotions after the mother’s behavior. Other than the F and T states, the BayesACT model also includes states X, actions A, and observations Ω . X represents all the things that the system needs to know about users’ behaviors and the system state. Actions A are things the system does to change the state or the human behaviors. Observations Ω are all the evidences that the system observes for finding the state X. In ACT, the identities are an actor and an object. BayesACT encodes the identities to “agent” (i.e. a machine) and “client” (i.e. a human) instead. The action of the agent is extended to $\mathbf{B} = \{A, \mathbf{B}_a\}$ to represent the action A and the emotional content of the action \mathbf{B}_a . For example, when the agent ignores the client, \mathbf{B}_a will be (-1.58, -0.75, -1.44). The client’s behavior is set in the fundamental sentiment F_b .

Here are the main formulas in BayesACT:

- $\varphi(\mathbf{f}', \boldsymbol{\tau}') \propto e^{-(\mathbf{f}' - \boldsymbol{\tau}')^T \Sigma^{-1} (\mathbf{f}' - \boldsymbol{\tau}')}$ denotes the deflection between fundamental sentiments \mathbf{F} and transient impressions \mathbf{T} .
- $Pr(\mathbf{x}' | \mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a)$ defines how the system progresses given the previous state, the fundamental and transient sentiments, and the action of the agent.
- $Pr(\boldsymbol{\omega}_f | \mathbf{f})$ and $Pr(\boldsymbol{\omega}_x | \mathbf{x})$ denote observation functions for the client behavior sentiment and the system state respectively.

BayesACT is a recursive Bayesian estimation model that it estimates and updates variables such as fundamental sentiments F and transient impressions T over time using

incoming actions A and observations Ω . BayesACT also represents emotions as probability distributions, and the emotion states can be calculated at each time based on the vector difference between fundamentals and transients [26].

4.2 BayesACT Simulation

In this work, we use BayesACT to simulate interactions. BayesACT simulates the interactions between a user and an avatar from the avatar’s perspective. In the simulation, the user and the avatar take turns to give actions. In each turn, in addition to providing an action, the user also needs to supply the current emotion state. The avatar perceives the user’s transient impressions and gains an understanding of the user’s fundamental sentiment. In this experiment, we supply the same conversations in the video from Semaine database as the input to BayesACT and analyze the learned users’ sentiment change for four avatars with different emotional characteristics.

We set the user’s identity to ‘student’, since all participants in the experiment were undergraduate or graduate students. In each turn, the user or the agent performs the action ‘talk to’ to the other party. When it is the user’s turn, the current (sampled every 5 seconds) EPA values from the facial expression database are supplied as the emotional signal. We present the simulations using the ground truth as input to more clearly show the emotional prediction mechanism at work. The raw EPA values ranged $[-1, 1]$ are transformed using a tangent function

$$x' = 2.77 * \tan(x) \tag{4.2}$$

to range $[-4.3, 4.3]$, which x is the original EPA and x' is the updated value. After each turn, the avatar generates two groups of EPA values: the avatar’s emotion and the user’s emotion. The EPA value of an individual’s emotion is calculated mainly on their action, but it is also updated by an observed value from their facial expressions. The portion based on actions and observations can be changed. In addition, the avatar converts the user’s and its own emotion states into labels. A label is an adjective chosen from a dictionary that has the shortest Euclidean distance from the user’s current emotion EPA. However, we use cosine distance as the distance measure when calculating labels. Compared to Euclidean distance that the original BayesACT uses, this metric better preserves the vector direction, when searching in a non-uniformly distributed label set.

4.3 Emotion Change in BayesACT Simulation

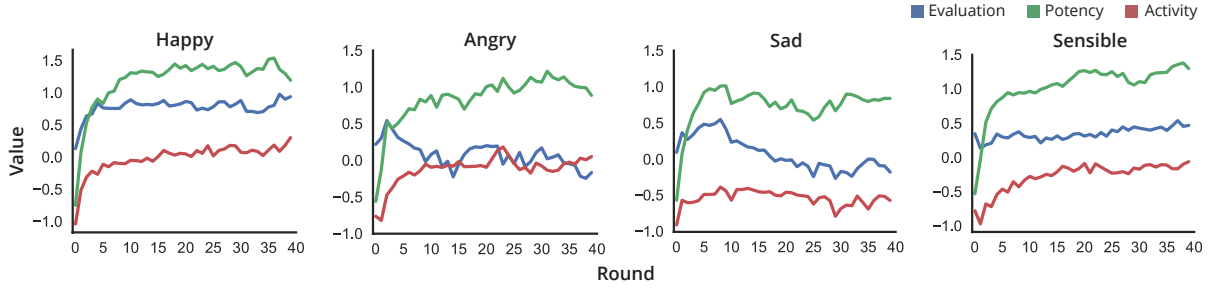


Figure 4.1: Avatars posterior estimate of the user’s emotions.

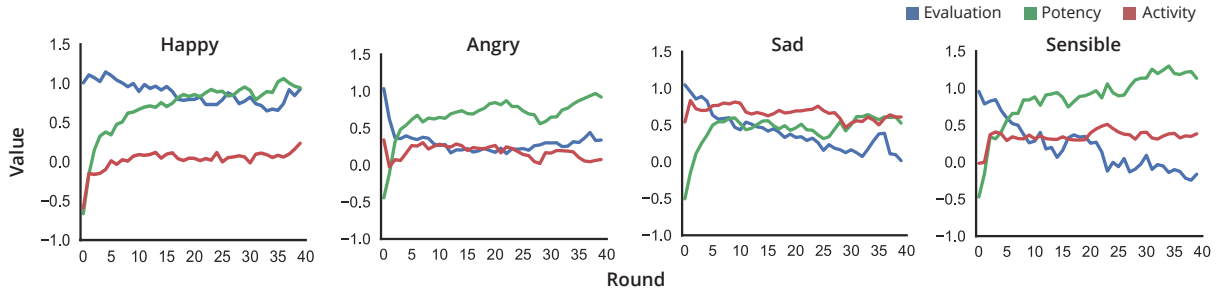


Figure 4.2: Avatars posterior estimate of the user’s transient impressions.

We first analyze the change of EPA values during the conversation using BayesACT simulations. The BayesACT simulations are expected to learn the user’s fundamental sentiment. We want to know given avatars with distinct emotion characteristics, how and to what extent avatars with distinct emotion characteristics affect the users emotional states. We use the same 18 video clips in the test set from Semaine database and categorize the clips to four types based on the avatar’s character. The number of simulations for each character is listed in Table 4.1. When aggregating the conversation simulations, the averaged value serves as a measurement of tendency towards greater valence, power, and activation. Since the lengths of the videos are varied, we align our data to use only the first 200 seconds of each conversation.

The experiment results are illustrated in Figure 4.1 and Figure 4.2, showing the avatars posterior estimate of the users emotions and transient impressions of the users identity, respectively. The figures illustrate the averaged values at each time stamp for Evaluation,

Table 4.1: Number of simulations for each type of avatar

	Happy	Angry	Sad	Sensible
Number of simulations	5	5	3	5

Potency, and Activity across all 18 test videos. The user’s emotion will change significantly when the avatar shows different characteristics to the user. This answers the question that the avatar does have an important effect on the user’s emotions. Both the avatar’s and the user’s emotions are proportional to the deflection which is the Euclidean distance between fundamental sentiments and transient impressions. A small deflection ensures the conversation running smoothly, but a large deflection makes humans feel uncomfortable which is they try to avoid. $Pr(\mathbf{e}'|\mathbf{f}, \boldsymbol{\tau}, \boldsymbol{\Omega}_e) \propto Pr(\boldsymbol{\Omega}_{e'}|\mathbf{e}')Pr(\mathbf{e}'|\mathbf{f}, \boldsymbol{\tau})$ denotes the updated emotion given the fundamental and transient sentiments, and observations on the emotion. A student is seen as [1.49, 0.31, 0.75], which is positive, less powerful, and a bit active. Users are confident and powerful when talking to a happy avatar, since EPA of the happy avatar’s behaviors is similar to what a student should behave. Therefore, the deflection is small and the conversation goes smoothly. However, when the avatar is angry, the avatar behaves unpleasant, much powerful, and active, which makes the users feel negative and powerless. Similarly, when the avatar is sad, the users maintain a positive attitude, but they are weak and less active. Moreover, when the avatar change to be sensible, the users feel less positive compared with talking to a happy avatar.

4.4 Sentiment Labels

To better visualize how different avatars affect the user’s emotion, we aggregate the sentiment labels generated from the BayesACT simulations, which approximate users’ emotion states. The aggregated labels are used to create four word clouds as shown in Figure 4.3¹. Each word cloud contains the top 15 words that describe the user’s emotions from the avatar’s perspective when interacting with different avatars. The size of each word is dictated by its frequency of appearance.

In the word clouds, the word ‘reverent’ appears the most number of times in all conditions, understandably representing the mental state of a student interacting with a person

¹We used <http://www.wordclouds.com/> for generating our word clouds.



Figure 4.3: Four word clouds created by the top 15 words that describe the user’s feeling when talking to Poppy (Happy), Spike (Angry), Obadiah (Sad), and Prudence (Sensible).

who leads the conversation with more or less greater extents of power. Other than ‘reverent’, the users show different emotions towards different types of avatars. From the avatar’s perspective, the users feel ‘infatuated’, ‘intelligent’, ‘wise’, ‘touched’, and ‘confident’ when talking to a happy person. They feel ‘strict’, ‘dogmatic’, and ‘authoritarian’, interacting with an angry avatar. When talking to a sad avatar, the users feel ‘sorry’, ‘touched’, ‘remorseful’, ‘repentant’, and ‘middle-aged’. When talking to a sensible avatar, the users feel ‘middle-aged’, ‘sly’, ‘intelligent’, and ‘touched’. The word cloud provides us with an impression on how the agent processes users’ emotions given their facial expressions in the BayesACT simulations.

Chapter 5

Game Integration

After the affect prediction, the next step is to integrate the facial expression recognition into a real prompting system and to see whether the system provides different prompts based on users' facial expressions. In this chapter, we implement our affect recognition module into an iterated prisoner's dilemma game, in which a user can play the prisoner's dilemma with a virtual human. The virtual human can change its facial expressions and actions based on users' choices and emotions. The iterated prisoner's game is also based on BayesACT and we use several test cases to ensure the affect recognition module works in the system.

5.1 Prisoner's Dilemma Game

The prisoner's dilemma is a standard example in game theory. Two individuals are imprisoned and each prisoner has only two options: betray the other or remain silent to cooperate with the other prisoner. There is no way for them to communicate before making a move.

Table 5.1: Payoff matrix in prisoner's dilemma

Feature Descriptor	Cooperation	Defection
Cooperation	-1, -1	0, -3
Defection	-3, 0	-2, -2

Table 5.2: Payoff matrix in the new iterated prisoner’s dilemma game

Feature Descriptor	Cooperation	Defection
Cooperation	2, 2	0, 3
Defection	3, 0	1, 1

As a result, there are four combinations in total and the payoff matrix is shown in Table 5.1. The results are:

- If both of them choose to cooperate, they will have to serve one year in prison, which gives the best result for them
- If one betrays the other but the other chooses to cooperate, the betrayed one will serve three years in prison
- If they both choose betray each other, they will have to serve two years in prison

For a single play, choosing cooperation gives the most benefit for both of them, but the dominant strategy is to choose defect which is the Nash equilibrium in the game. The iterated prisoner’s dilemma game has the same rule, but two players can continue playing the game for multiple rounds, which makes the results more complicated. The iterated game uses virtual coins as a motivation and the goal for each player is to collect as many coins as possible. The special part of the game is that the user plays with a virtual human and the virtual human shows its facial expressions and says its emotions in each round.

As the game begins, the virtual human introduces the rule at first. In each turn, both the user and the virtual human can choose either “give 2” as a cooperative action or “take 1” as a defection. When the user clicks a button, the virtual human cannot see user’s decision until it makes its decision. The virtual human’s move is selected by BayesACT. After both of them have made the choice, the result will be shown and the players will earn coins based on their move. The payoff matrix is updated in Table 5.2. After each round, BayesACT calculates the virtual human’s next move and current EPA value based on previous information. When the result is shown, the virtual human will show its action, say a selected sentence to declare its emotion, and change its facial expression based on her current EPA value. For example, if the user always chooses “take 1” to defect, the virtual human will get a negative EPA value and it will choose to defect as well. At the same time,

it will show an angry look and say “You are making me frustrated.” The sentence dictionary contains about a hundred sentences such as “I am starting to get angry”, “Thanks for being friendly”, and “I am a little uneasy”. The sentences and the emotional labels from the modifier dictionary are embedded using word2vec in advance, and each emotion label is mapped to the closest sentence using cosine similarity. Therefore, in each round, a sentence is selected that has the maximum cosine similarity from the virtual human’s current EPA. Both of players’ scores and the virtual human’s last move are shown on the screen. The game interface is shown in Figure 5.2 and Figure 5.3. The virtual human is created from the Interactive Assistance Lab at University of Colorado at Boulder [63].

5.2 System Design

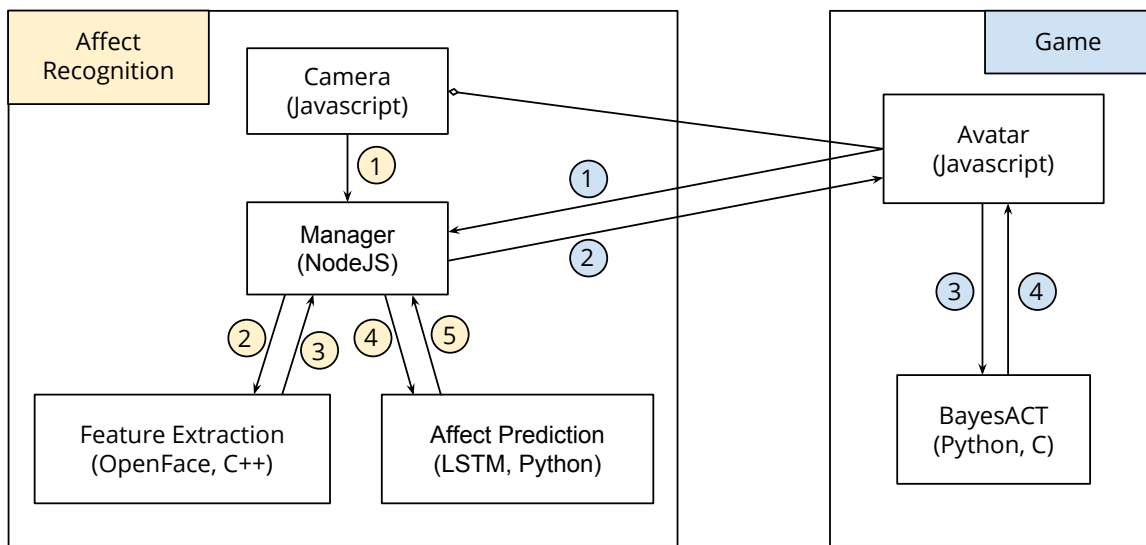


Figure 5.1: An overview of the integration system

The iterated prisoner’s dilemma game is a web game and it was implemented by an undergraduate research assistant as a research project. The structure of the previous project involves a front end virtual human with a game interface and a back end to process and select the next move for the virtual human. For the front end interface, it is written in JavaScript. The interface shows current scores for two players and it takes the user’s next move to the back end. The back end is written in Python and C. It processes the

user’s action and uses BayesACT to calculate the virtual human’s emotion and its next action. After the decision is made, the back end sends the results back to the front end. The results include the virtual human’s facial expressions, next move, and an emotion label based on the calculated EPA value.

In our work, we integrate the affect recognition module into the system. An overview of the integrated system is shown in Figure 5.1. At first, we turn on a camera to record the user’s facial expressions and the real-time tracking is shown on the bottom right of the game interface. We use an open source library JpegCamera [65] to take one picture of the user every five seconds and upload the images. Then we use a short program in NodeJS to manage the image process. The program saves the images received from the previous step in a local folder with a time-stamp included. The program is also responsible for cleaning old images which are created more than one minute before.

We use another program in NodeJS as a manager to control the whole recognition process. It first sends images to a server to extract feature descriptors. Three feature descriptors (AU, HOG, FHOG) are extracted in Chapter 3, but we only consider FHOG in the system, since FHOG gives the best prediction result. We modify one of the OpenFace programs so that the program can process an HTTP request to extract FHOG features from any received image and return the feature descriptor to the manager. After the feature extraction, the program also uses PCA to reduce the feature dimension from 4464 to 153. After the FHOG feature is returned, the manager sends another request to affect prediction, which is another server to predict EPA values using LSTM. LSTM is selected because of its lower RMSE compared with SVR. Since the EPA values from Semaine database are all measured from [-1, 1] but the range of EPA in ACT is from [-4.3, 4.3], we use a tangent function

$$x' = 2.77 * \tan(x) \tag{5.1}$$

to update the range, which x is the original EPA and x' is the updated value. The reason we use a tangent instead of a linear scale is to avoid increasing the predicted error after transformation. After the affect prediction server gets the EPA score, it will send back the results to the manager after the process finishes and all the EPA values will be stored in the manager.

When the user clicks on “give 2” or “take 1” as a next move, the front end in the game will immediately send a request to the manager to retrieve EPA values of the user’s facial expressions. Then the manager will calculate the most recent EPA and clear all the values after it delivers the result. We can also change the program to calculate the averaged EPA values from the previous move to the current move, but it requires the user to keep an facial expression for a few seconds. Therefore, we think using the most recent value is better.

After retrieving the EPA of the user’s facial expressions, the front end virtual human sends both its next action and current emotion to the back end. BayesACT calculates the virtual human’s information based on the user’s move, facial expressions, and previous records. In the end, BayesACT sends the virtual human’s move and facial expressions to the front end. The interface updates the players’ actions, current coins, and the virtual human’s emotions in order. Then the game continues to the next round. The game is extended to a new version in case the user does not want to enable emotion detection during the game. Three options show up in the beginning of the game: no emotion, emoji emotions, and facial emotion recognition. When the first option is chosen, the camera is hidden and $[0, 0, 0]$ is sent as EPA values to the virtual human after each move. If the user only wants to send basic emotions, the emoji option will list six emojis to represent the six basic emotions (happy, sad, surprised, angry, disgusted, and fearful). Users can choose one emoji to represent their emotions and the selected emoji will be converted to an EPA value. The third option enables the affect recognition and analyses the participant’s facial expressions as emotions. Since it is hard to measure and compare those three conditions quantitatively, we will only test the third option in our studies.

5.3 Tests and Discussions

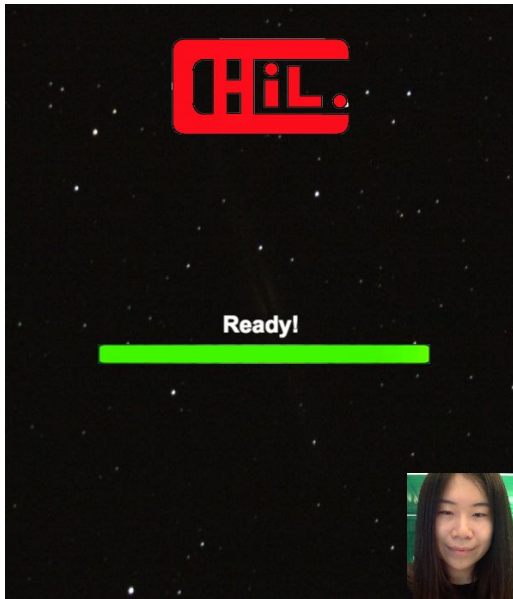
We invite four students in the Computational Health Informatics Lab to test the game informally on the updated system. During the game, we record the game interface and participants’ facial expressions for later review. We introduce the game first and then start recording their performance. Participants can use different strategies to play and they can stop whenever they want. The four students play 83 rounds in total, and each of them plays about 20 rounds. We only test the system informally in this thesis to ensure users feel entertained to play the game with facial recognition installed.

The new system with enabled emotion detection provides a different gaming experience in which the virtual human shows different responses from its strategies, facial expressions, and responding sentences based on users’ moves and emotions. The virtual human has different strategies to play the game with users’ different emotions. We surprisingly find the virtual human defects more if the user shows an angry face (50% of the time) than a happy facial expression (36%). Because an angry player has more probability to defect, the virtual human chooses to defect as well. The change of the virtual human’s facial expressions is also noticeable. If the user defects and is angry, the virtual human will show sad or angry facial expressions most of the time (83%); when the user defects but is happy, the virtual human will have less possibilities to provide negative emotions (70%). However,

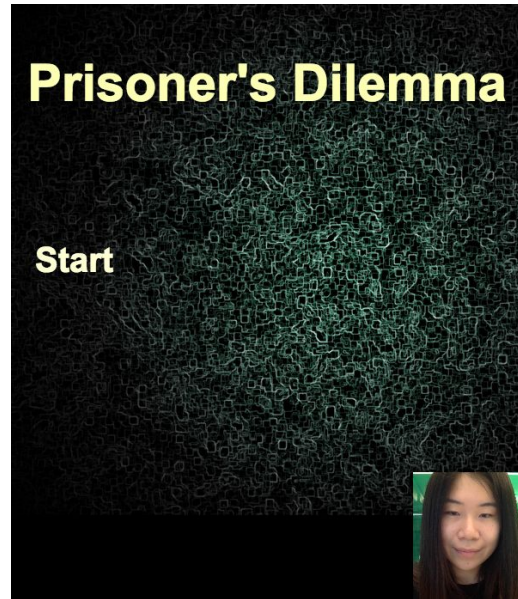
the virtual human’s facial expressions will gradually appear angry when the user continues defecting. In addition, the virtual human says different sentences to reflect its emotions based on the user’s moves and emotions. We list a few examples we find interesting during tests. The screenshots of these examples are shown in Figure 5.4:

- (a) When a participant showed a smiling face but chose “take 1”, the system recognized the facial expression as “playful” and the virtual human chose to cooperate. The virtual human said “I feel like a fool” when the virtual human knew the user’s move since it did not expect the user was going to defect.
- (b) The virtual human said “You are making me upset” even though the participant chose to cooperate but with a disgusted face, which demonstrates the user’s facial expressions also affect the game.
- (c) When the user chose “give 2” with a smiling face, the virtual human said “That was cute” with a happy facial expression.
- (d) When the user cooperated with a happy face but the virtual human said “I am starting to get angry” and showed a neutral facial expression, which is a bit strange here.

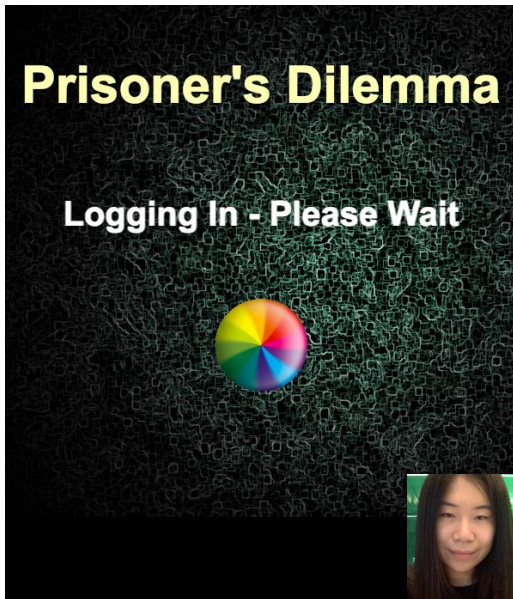
We find that users obtain more personal responses by integrating affect recognition into the system and both users’ moves and facial expressions have an effect on the virtual human’s side. Some responses from the virtual human are smart and entertaining, but a few responses are awkward and not accurate. For example, the virtual human feels sad even though the user cooperates with a smiling face. In this study, we only invite a few participants to test the game informally. A more comprehensive user study could be further conducted to reveal the efficacy of intelligent agents and elicit user feedback.



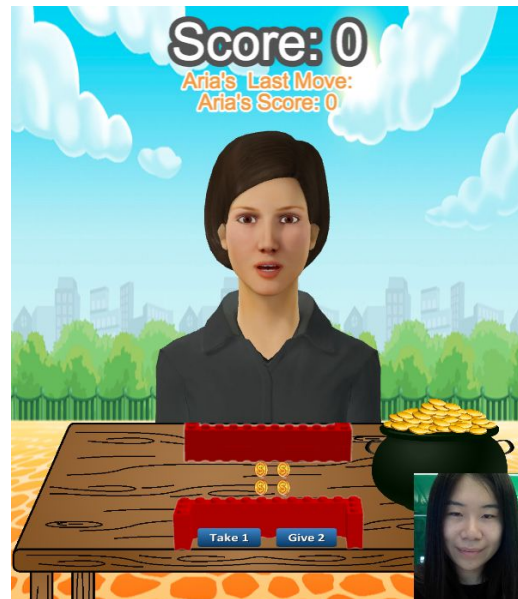
(a) game loading



(b) start a new game



(c) game log in



(d) the avatar introduces the prisoner's dilemma game

Figure 5.2: The game interface of the iterated prisoner's dilemma game with camera enabled



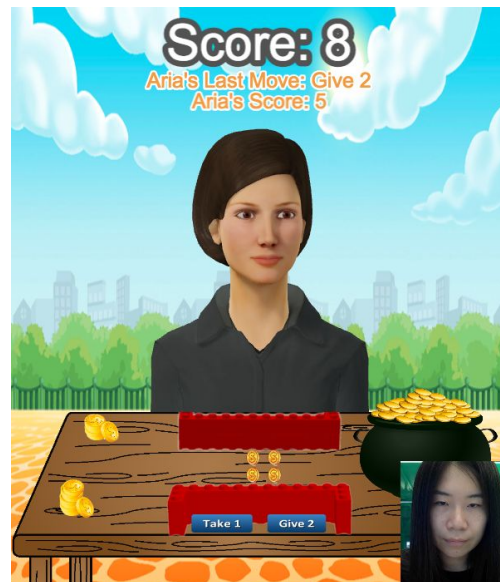
(a) a prompt indicates the avatar is still thinking for the next move



(b) the avatar is happy when the user chooses to cooperate

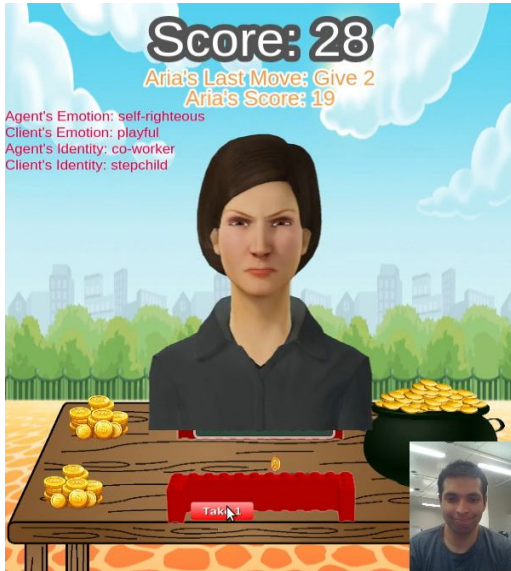


(c) the avatar is sad when it loses coins



(d) the avatar moves its face to wait for the user's next action

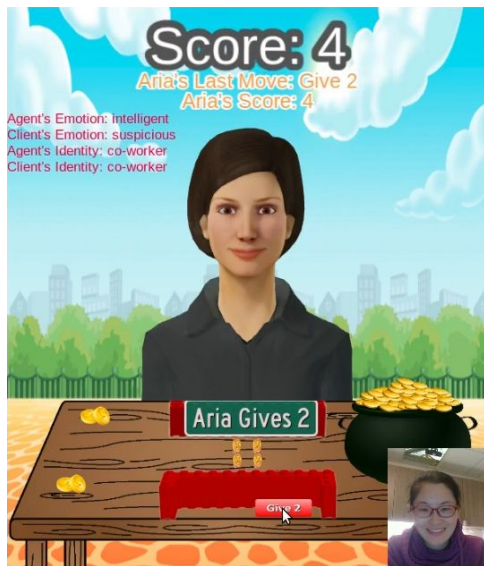
Figure 5.3: An overview of the game interface and the facial expressions on the virtual human



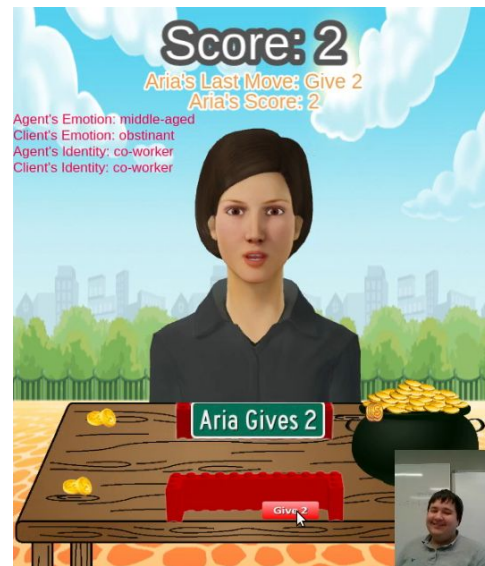
(a) The virtual human said “I feel like a fool”



(b) The virtual human said “You are making me upset”



(c) The virtual human said “That was cute”



(d) The virtual human said “I am starting to get angry”

Figure 5.4: Examples of participants playing the game

Chapter 6

Conclusion

In this thesis, we build an automatic affect recognizer to predict user emotions in the Evaluation-Potency-Activity (EPA) space. We compare three different feature descriptors (AU, HOG, FHOG) and build models with two machine learning approaches (SVM-Regression and RNN-BLSTM). We test the models with the Semaine database that provides annotated EPA values of human natural expressions when interacting with avatars.

This thesis further demonstrates that affective information of users can be perceived and thus can be used to build affective prompting systems. In the simulations, we assume the identity of user to be student and action as talk to in all the cases. However, this might not always hold true. Despite these assumptions we see some distinctive words generated in the word cloud based on the behaviour of different avatars. We present BayesACT simulation results using the database labels as input, but clearly it will be preferable to use the automated facial expression recognition. Our analysis shows that improvements will be needed in continuous facial expression recognition in order to make this feasible.

Finally, we integrate our recognizer into a prisoner's dilemma game in which a user can play with a virtual human. The virtual human replies with different facial expressions and emotions based on the user's moves and facial emotions. The game provides a more enriched play experience to users. The realistic responses from the virtual human make users feel they are not playing with a machine. The work also demonstrates the potential of further integrating emotion recognition into other prompting systems.

However, integrating affect recognition into game systems may have less effect than integrating into tutoring systems or assistive systems. Users can choose to show faked facial expressions to the virtual human among the game as a strategy. For example, users can show an angry face even though they are happy to push the virtual human to be

more cooperative. The faked expressions may give them more possibilities to win, so the virtual human cannot trust the EPA values from users. In a tutoring system or an assistive system, the EPA values of the user’s facial expressions are more reliable. Therefore, the systems that help people to learn or to complete a task have more advantages when facial expression detection is enabled.

We should note that this thesis is only a small step towards building emotionally intelligent prompting systems. The accuracy on affect recognition still can be improved from two aspects: the dataset as the ground truth values and the feature descriptors with modeling approaches. In our studies, we only calculate the averaged values among all the raters as the ground truth. However, the variance of the raters in the dataset would cause noise in the baseline measurements. Therefore, doing an analysis on the rating variance may help to quantify the noise and provide a better way to combine annotators’ ratings. On the other hand, we investigate a number of feature descriptors and machine learning approaches, but the space for finding the optimal approach is huge. For example, in addition to SVM-Regression and RNN-BLSTM that we have tested in the thesis, we might also explore building a high-performance Convolutional Neural Network (CNN) to extract feature descriptors with a recurrent neural network to build models. Also, we could leverage other visual cues such as hand movement and audio signals like voice frequency to improve the robustness of our recognition technique. Improving the performance on affect recognition provides a fundamental step to further validate the models on applications such as the BayesACT simulations and the integrated game we used in our studies.

In this thesis, we only informally test the prisoner’s dilemma game. A more comprehensive user study could be further conducted to reveal the efficacy of intelligent agents and elicit user feedback. Moreover, other prompting systems can be tested by integrating the affect recognizer into them. For example, a hand-washing system with an integrated BayesACT model, uses a virtual assistant to provide instructions to help people with Alzheimer’s independently finish a hand-washing task [33]. The automatic affect recognition can also be integrated into the hand-washing system so that the virtual assistant can provide more tailored prompts based on users’ behaviors and emotional states.

References

- [1] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.
- [2] Nalini Ambady and Robert Rosenthal. *Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis*. American Psychological Association, 1992.
- [3] T. Baltruaitis, P. Robinson, and L. P. Morency. OpenFace: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, March 2016.
- [4] Tadas Baltruaitis, Ntombikayise Banda, and Peter Robinson. Dimensional affect recognition using continuous conditional random fields. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE, 2013.
- [5] Carlos Busso, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann, and Shrikanth Narayanan. Analysis of Emotion Recognition Using Facial Expressions, Speech and Multimodal Information. In *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI '04*, pages 205–211, New York, NY, USA, 2004. ACM.
- [6] Tanja Bnziger and Klaus R. Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. *Blueprint for affective computing: A sourcebook*, pages 271–294, 2010.
- [7] Maureen Caudill. Neural networks primer, part I. *AI expert*, 2(12):46–52, 1987.
- [8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

- [9] Francois Chollet. keras. <https://github.com/fchollet/keras>, 2015.
- [10] Jeffrey F. Cohn and Karen L. Schmidt. The timing of facial motion in posed and spontaneous smiles. *International Journal of Wavelets, Multiresolution and Information Processing*, 2(02):121–132, 2004.
- [11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [12] Shane F. Cotter. Sparse representation for accurate classification of corrupted and occluded facial expressions. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 838–841. IEEE, 2010.
- [13] Roddy Cowie, Ellen Douglas-Cowie, Susie Savvidou*, Edelle McMahon, Martin Sawey, and Marc Schrder. 'FEELTRACE': An instrument for recording perceived emotion in real time. In *ISCA tutorial and research workshop (ITRW) on speech and emotion*, 2000.
- [14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, volume 1, pages 886–893 vol. 1, June 2005.
- [15] S. D'Mello, R. W. Picard, and A. Graesser. Toward an Affect-Sensitive AutoTutor. *IEEE Intelligent Systems*, 22(4):53–61, July 2007.
- [16] Paul Ekman. Are there basic emotions? *Psychological Review*, 99(3):550–553, 1992.
- [17] Paul Ekman and Wallace V. Friesen. *Manual for the facial action coding system*. Consulting Psychologists Press, 1978.
- [18] Florian Eyben, Martin Wllmer, Tony Poitschke, Bjrjn Schuller, Christoph Blaschke, Fä, Berthold Rber, and Nhu Nguyen-Thien. Emotion on the RoadNecessity, Acceptance, and Feasibility of Affective Computing in the Car. *Advances in Human-Computer Interaction*, 2010:e263593, July 2010.
- [19] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.
- [20] Johnny RJ Fontaine, Klaus R. Scherer, Etienne B. Roesch, and Phoebe C. Ellsworth. The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050–1057, 2007.

- [21] Yoav Freund and Robert Schapire. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.
- [22] Wallace V. Friesen and Paul Ekman. EMFACS-7: Emotional facial action coding system. *Unpublished manuscript, University of California at San Francisco*, 2(36):1, 1983.
- [23] Alex Graves and Jrgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.
- [24] Hatice Gunes and Bjrn Schuller. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image and Vision Computing*, 31(2):120–136, February 2013.
- [25] J. A. Healey and R. W. Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, June 2005.
- [26] David R Heise. *Expressive order: Confirming sentiments in social actions*. Springer Science & Business Media, 2007.
- [27] David R. Heise. *Surveying cultures: Discovering shared conceptions and sentiments*. John Wiley & Sons, 2010.
- [28] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [29] Jesse Hoey, Tobias Schroder, and Areej Alhothali. Bayesian affect control theory. In *ACII, 2013 Humaine Association Conference on*, pages 166–172. IEEE, 2013.
- [30] Jesse Hoey, Tobias Schrder, and Areej Alhothali. Affect control processes: Intelligent affective interaction using a partially observable Markov decision process. *Artificial Intelligence*, 230:134–172, 2016.
- [31] Itseez. Open source computer vision library. <https://github.com/itseez/opencv>, 2015.
- [32] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

- [33] Luyuan Lin, Stephen Czarnuch, Aarti Malhotra, Lifei Yu, Tobias Schrder, and Jesse Hoey. Affectively aligned cognitive assistance using Bayesian affect control theory. In *International Workshop on Ambient Assisted Living*, pages 279–287. Springer, 2014.
- [34] David G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [35] P. Lucey, J. F. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K. M. Prkachin. Automatically Detecting Pain in Video Through Facial Action Units. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(3):664–674, June 2011.
- [36] 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 *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.
- [37] Mohammad H. Mahoor, Mu Zhou, Kevin L. Veon, S. Mohammad Mavadati, and Jeffrey F. Cohn. Facial action unit recognition with sparse representation. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 336–342. IEEE, 2011.
- [38] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. 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, January 2012.
- [39] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. *Current Psychology*, 14(4):261–292, December 1996.
- [40] Michael T. Motley and Carl T. Camden. Facial expression of emotion: A comparison of posed expressions versus spontaneous expressions in an interpersonal communication setting. *Western Journal of Speech Communication*, 52(1):1–22, April 1988.
- [41] M. A. Nicolaou, H. Gunes, and M. Pantic. Continuous Prediction of Spontaneous Affect from Multiple Cues and Modalities in Valence-Arousal Space. *IEEE Transactions on Affective Computing*, 2(2):92–105, April 2011.

- [42] Celia J. Orona. Temporality and identity loss due to Alzheimer’s disease. *Social Science & Medicine*, 30(11):1247–1256, 1990.
- [43] Charles E. Osgood, G. J. Suci, and Percy H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, Urbana, 1957.
- [44] Charles Egerton Osgood, William H. May, and Murray S. Miron. *Cross-cultural universals of affective meaning*. University of Illinois Press, 1975.
- [45] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 5–pp. IEEE, 2005.
- [46] Dawn T. Robinson, Lynn Smith-Lovin, and Allison K. Wisecup. Affect control theory. In *Handbook of the sociology of emotions*, pages 179–202. Springer, 2006.
- [47] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.
- [48] J. Sanghvi, G. Castellano, I. Leite, A. Pereira, P. W. McOwan, and A. Paiva. Automatic analysis of affective postures and body motion to detect engagement with a game companion. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 305–311, March 2011.
- [49] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, June 2015.
- [50] Nikolaos Savva and Nadia Bianchi-Berthouze. Automatic Recognition of Affective Body Movement in a Video Game Scenario. In *Intelligent Technologies for Interactive Entertainment*, pages 149–159. Springer, Berlin, Heidelberg, May 2011.
- [51] Bjrn Schuller, Michel Valstar, Florian Eyben, Gary McKeown, Roddy Cowie, and Maja Pantic. Avec 2011the first international audio/visual emotion challenge. *Affective Computing and Intelligent Interaction*, pages 415–424, 2011.
- [52] Bjrn Schuller, Michel Valster, Florian Eyben, Roddy Cowie, and Maja Pantic. AVEC 2012: the continuous audio/visual emotion challenge. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 449–456. ACM, 2012.

- [53] L. C. De Silva, T. Miyasato, and R. Nakatsu. Facial emotion recognition using multi-modal information. In *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat., volume 1*, pages 397–401 vol.1, September 1997.
- [54] Jianhua Tao and Tieniu Tan. Affective computing: A review. In *International Conference on Affective computing and intelligent interaction*, pages 981–995. Springer, 2005.
- [55] Y. I. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, February 2001.
- [56] Michel Valstar, Jonathan Gratch, Bjrn Schuller, Fabien Ringeval, Dennis Lalanne, Mercedes Torres Torres, Stefan Scherer, Giota Stratou, Roddy Cowie, and Maja Pantic. AVEC 2016: Depression, Mood, and Emotion Recognition Workshop and Challenge. In *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM, 2016.
- [57] Michel Valstar, Bjrn Schuller, Kirsty Smith, Timur Almaev, Florian Eyben, Jarek Krajewski, Roddy Cowie, and Maja Pantic. AVEC 2014: 3d dimensional affect and depression recognition challenge. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM, 2014.
- [58] Michel Valstar, Bjrn Schuller, Kirsty Smith, Florian Eyben, Bihan Jiang, Sanjay Birlakha, Sebastian Schnieder, Roddy Cowie, and Maja Pantic. AVEC 2013: the continuous audio/visual emotion and depression recognition challenge. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 3–10. ACM, 2013.
- [59] Michel F. Valstar, Timur Almaev, Jeffrey M. Girard, Gary McKeown, Marc Mehu, Lijun Yin, Maja Pantic, and Jeffrey F. Cohn. Fera 2015-second facial expression recognition and analysis challenge. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–8. IEEE, 2015.
- [60] Michel F. Valstar, Bihan Jiang, Marc Mehu, Maja Pantic, and Klaus Scherer. The first facial expression recognition and analysis challenge. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 921–926. IEEE, 2011.

- [61] Michel F. Valstar, Maja Pantic, Zara Ambadar, and Jeffrey F. Cohn. Spontaneous vs. Posed Facial Behavior: Automatic Analysis of Brow Actions. In *Proceedings of the 8th International Conference on Multimodal Interfaces, ICMI '06*, pages 162–170, New York, NY, USA, 2006. ACM.
- [62] Michel F. Valstar, Enrique Snchez-Lozano, Jeffrey F. Cohn, Lszl A. Jeni, Jeffrey M. Girard, Zheng Zhang, Lijun Yin, and Maja Pantic. FERA 2017-Addressing Head Pose in the Third Facial Expression Recognition and Analysis Challenge. *arXiv preprint arXiv:1702.04174*, 2017.
- [63] 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.
- [64] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [65] Adam Wrbel. Jpegcamera. https://github.com/amw/jpeg_camera, 2016.
- [66] M. Yang, L. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *CVPR 2011*, pages 625–632, June 2011.
- [67] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato. A 3d facial expression database for facial behavior research. In *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*, pages 211–216. IEEE, 2006.
- [68] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, January 2009.