UNIVERSITY OF MIAMI

CSC 794 REPORT

USER EMOTION ANALYSIS AND DEPRESSION RECOGNITION SYSTEM BASED ON HUMAN-COMPUTER INTERACTION

By
Zihao Wang

Supervisor:
Professor Dr. Liang Liang

Coral Gables, Florida

May 2020

Abstract

With the development of graphic human-computer interaction (HCI) agents, there is an explosive need for user experiences satisfaction. In many HCI systems, the general architecture includes 2 parts. One is a real-time multimodal agent of human-computer interaction, which enhances user experiences while interacting with the computer. The other is the voice feedback agent. Analysis of user experiences is an important research topic of HCI. Users' emotions and depression analysis is the most important and challenging problem in user experience analysis.

Towards this challenge, in this research, we proposed machine learning approaches to analyze user depression based on the HCI videos, and we have achieved promising results. Specifically, we proposed two methods for depression analysis. The first one is using continuous 3-dimension emotion representation (arousal, dominance and valence) and the Long Short-term Memory neural network (LSTM). It reaches the accuracy of 58% percent The second one is using discrete emotion features of the LSTM and k-nearest neighbor (KNN) classifier. The LSTM reaches the accuracy of 56% and the KNN accuracy is 70% that is significantly higher than the others above. We took three steps in the research: (1) implement a facial expression recognition system based on the VGG19 neural network to recognize facial emotion in each HCI video frame; (2) generate a feature vector of length 512 for each face in each frame by using the output of the middle of VGG19; (3) use machine learning methods to predict the depression level based on the 3-dimension emotion representation and discrete emotion features.

All the proposed machine learning models are trained and evaluated on the Audio-Visual Emotion recognition Challenge (AVEC) 2014 dataset. The experimental results

show that the most effective method is the K nearest neighbor (KNN) using the discrete emotion feature. The accuracy reaches 70% and the recall for non-depression detection reaches 96.8% which is higher than the baseline 85.9%. The result illustrates it's a promising method and could be applied to real-world applications.

Keywords: emotions, depression, human-computer interaction, user experiences, machine learning

Table of Contents

1. Introduction

1.1. Motivation

The mental health problem, which affects humans during their life, was studied in the European Union Green Papers of 2005 [1] and 2008 [2]. Compare to other illnesses, mental illness influence people of ages, causing significant losses and stress to the economic environment, also the social, educational and justice fields.

To solve this problem, affective computing and social signal processing are two significant approaches. Affective computing is the science of emotion aware technology of automated emotion analysis and behavior expression. Social signal processing deals with the verbal and nonverbal signals during social interaction. Depression is correlated with social interaction and can be analyzed by affective computing because mood disorder is related to the affective state [3].

Based on the above mentioned phenomenon, in this research, we propose to develop a system that can detect the human depression degree during human-computer interaction. The research has a potential impact in the areas of psychological, social sciences and computer science.

1.2. Report outline

This report is organized as follows. Chapter 2 reviews the technologies of affective computing and machine learning. Chapter 3 introduces the architecture of the HCI system. Chapter 4 describes the method of predicting the continuous affective dimensions from each video frame. Chapter 5 presents the LSTM neural network for predicting depression degree by using continuous affective dimension value. Chapter 6 introduces methods of predicting depression value by using discrete emotional features. The report is concluded in Chapter 7, which also proposes some future work directions.

The process of this research is illustrated in Figure 1.1.

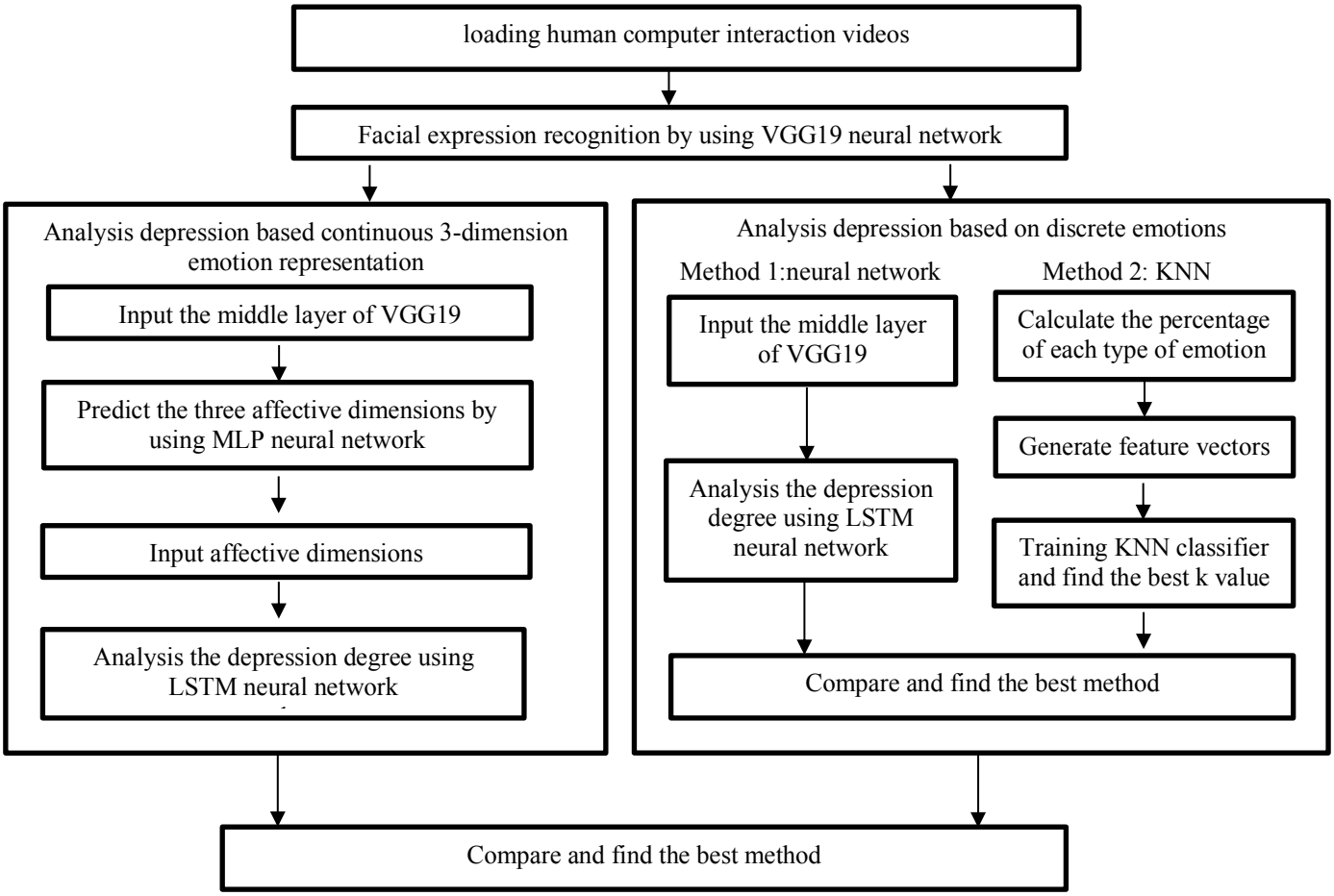


Figure 1.1: blueprint of the research

2. Literature review

2.1. Human-Computer interaction intelligent system

To analysis users' emotion and depression, video datastes were obtained from HCI intelligent systems [4, 5, 6, 7]. The system has 3 parts [8]. The first part is the sensor, and in this research, the sensor is the camera that records the human facial motion during the human-computer interaction process. The second part is the program that uses input information to calculate and output signals. The third part is the actuator which decodes output signals and give human readable feedback such as images, texts and audios displayed on the screen. A general system structure is shown in Figure 2.1.

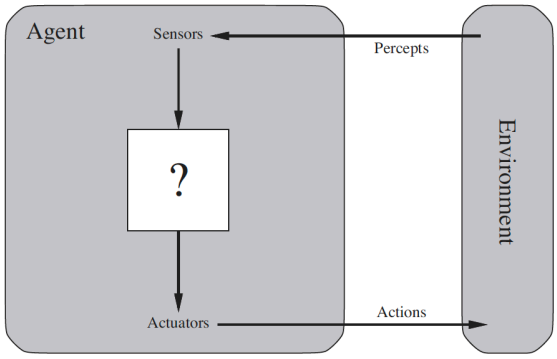
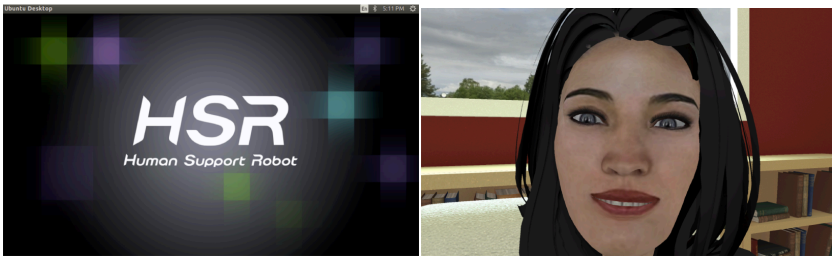


Figure 2.1: a general agent interact with environments through sensors and actuators [8]

There are 2 kinds of HCI systems, The one is foundation system that human interact with a computer with a Power Point guide frame, webcam and microphone [9].

The other is graphical user interface (GUI) system that is embedded into robot with special software for human to interact with [10]. One of the typical GUI systems is empathic embodied virtual agent(eEVA), that is embedded into Toyota HSR robot to enhance user experiences [11]. The default HSR interaction interface and the eEVA interface is illustrated in Figure 2.2.



(a) Default HSR interface (b) eEVA visual screen

Figure 2.2: Human-robot interface [11]

There are 2 parts of the eEVA system: modules and resource generic types. The basic idea of a module completely achieves a basic exactly functionality of the whole system. A module is decided by the task and the resources that be provided for solving the problem. A model get input that the resource needed and output a result that is processed by the resource. A model includes sensors that only provide resources, processors and effectors which require resources but not produce new data for the system use.

The modules are used for eEVA are listed in Table 2.1

Table 2.1: list of eEVA current modules [11]

Ref. N°	Type	Short name	Function description
1	Sensor	ChromeSpeech	Speech recognition using Google Chrome API
2	Processor	HapCharacter	Virtual character controller (body and face)
3	Processor	UserChoice	User interface for interacting with eEVA
4	Processor	WinSAPISynth	Speech synthesis using Windows SAPI
5	Effector	WebGLScene	Default 3D scene rendering
6	Effector	ROSHandler	ROS Communication through roslibjs

In eEVA, the most useful sensor is ChromeSpeech module which uses Google Speech API to recognize speech from the user by using the head microphone of HSR. It uses WinSAPISynth as a processor to generate speech. The communication between human and robot is using ROS Handler effector.

2.2. Facial emotion analysis

Facial expression analysis is essential to human-computer interaction [12] and is important in social interaction as well as social intelligence [13]. There are two representation models for facial expressions. one is categorical; the other is continuous [14].

2.2.1. Categorical model

The categorical model plays a more critical role than the continuous model in emotion classification, especially in stastics image classification. The categorical model tags each facial expression image into one of the seven prototypical expressions which are neutral, angry, disgust, fear, happy, sad and surprise. Peoples' appearance also influences the human expression classification [15].

Categorical facial expression analysis has two parts. The first part is the facial feature extraction. The second part is the facial expression classification.

For facial feature extraction, Gabor coding is used to represent facial expressions [16]. To get the Gabor code similarity space, feature maps that have the same spatial frequency and orientation preference were compared at corresponding points by using the normalized dot product. The 34-node grid is used to represent facial geometry (Figure 2.3). The similarity of one image grid and the other are calculated as average points overall the corresponding images.

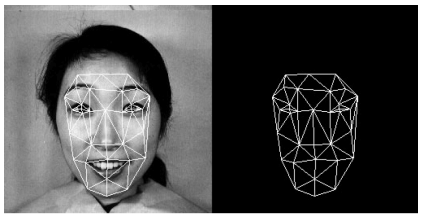


Figure 2.3: the 34 node grid used to represent facial geometry [16]

To extract the local information of facial images, facial images are divided into an equal number of blocks [14]. A histogram is calculated for each of the blocks. These histograms

are normalized and concentrated together to form the Accumulated Motion Descriptor (AMD) (Figure 2.4).

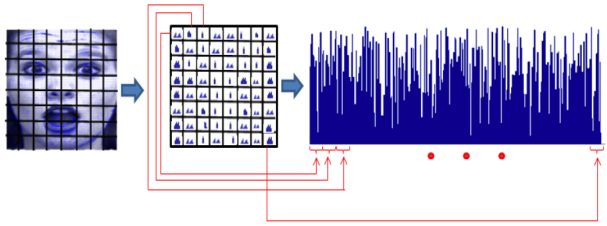


Figure 2.4: partitioning and accumulated histograms [14]

For image feature classification, there are some classical CNNs, such as the VGG19 used in this research [17]. The VGG19 is designed for large scale image recognition by increasing the depth of the neural network using small (3x3) convolution filters. VGG19 has achieved an accuracy of 70% for facial expression recognition [18].

2.2.2. Continuous model

Continuous model is better than the categorical model in time serious emotion analysis. Continuous models represent each emotion as a vector in 3 dimensions continuous space by using 3 affective dimensions of arousal, dominance and valence [19]. Arousal is the individual's global feeling of dynamism or lethargy. Dominance is an individual's sense of how much he/she feels to be in control of their situation. Valence describes the person rated feels positive or negative about the things.

The 3 affection dimensions were measured by the psychological experiments of human-computer interaction using the FEELTRACE software [20]. In theory, an emotion can be determined by a combination of the 3-affection dimension value. The interface of the FEELTRACE is illustrated in Figure 2.5 from [20].

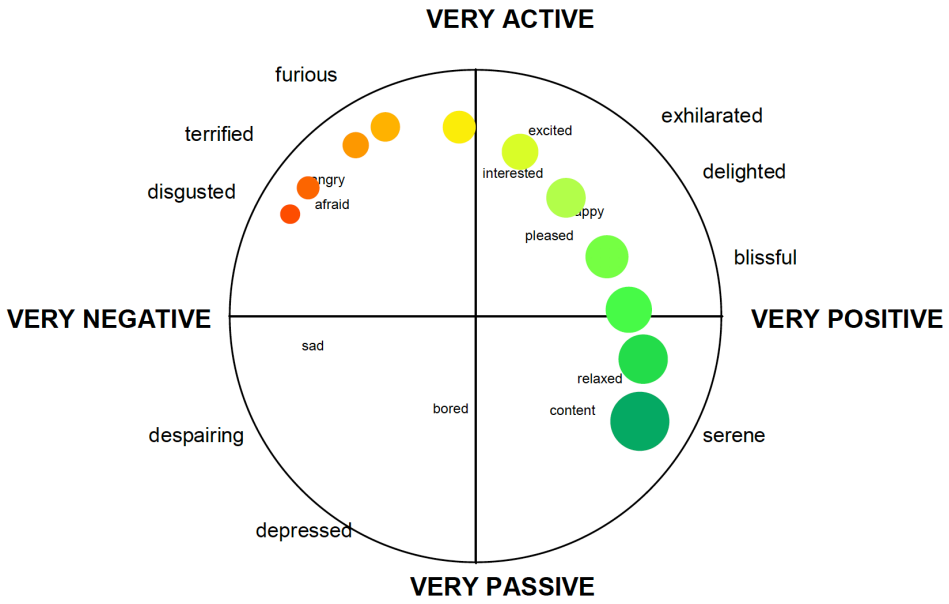


Figure 2.5: example of a FEELTRACE display during a tracking session. Cursor color changes from red/orange at the left-hand end of the arc, to yellow beside the active/passive axis, to bright green on the negative/positive axis, to blue-green at the right-hand end of the arc [20]

2.3. Depression degree analysis

In the 2014 Audio-Visual Emotion Challenge dataset [19], the depression degree is detected by 2 tasks which are named Northwind and Freeform. In the Northwind task, users say aloud an excerpt of the fable "Die Sonne und der Wind" (The North Wind and the Sun), spoken in the German language. And in the Freeform task, Participants respond to a number of questions such as: "What is your favorite dish?"; "What was your best gift, and why?"; "Discuss a sad childhood memory".

During the 2 tasks, there are 5 people who are responsible for measuring the 3 affection dimensions of each recorded video frame. The 3 affection dimensions are normalized to [-1,1] in order to analyze the depression.

The degree of depression is labeled for the whole video. The depression degree is an integer obtained from the Beck Depression Inventory-II (BDI-II) [21] which contains 21 questions. Each question is a forced-choice question scored from 0 to 3. The final value is

between 0 and 63. The meaning of the range is as follows: 0-13 means no depression, 14-19 indicates mild depression, 20-28 represents moderate depression, and 29-63 interprets high depression.

In the research based on the 2016 Audio-Visual Emotion Challenge dataset [22], the depression value is calculated by using PHQ-8 scores. According to [23], men and women have a different reaction to depression. The author proposed 2 separate decision trees analysis based on several features for man and woman without 3 affective dimension values. The 2 decision trees' structure is shown in Figure 2.6 and Figure 2.7. In the leave nodes, 0 means no depression and 1 means depression.

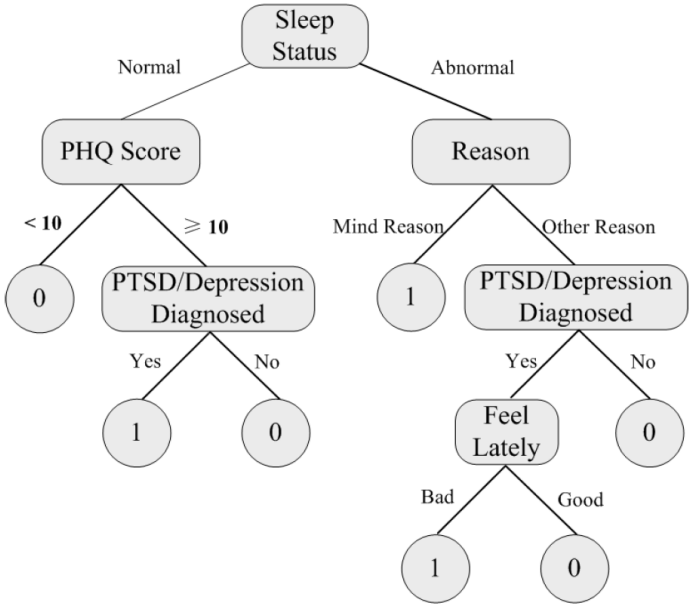


Figure 2.6: decision tree for the woman [22]

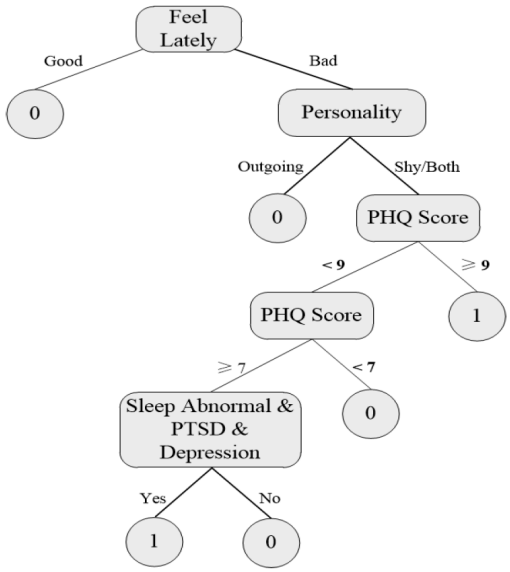


Figure 2.7: decision tree for the man [22]

2.4. Machine learning techniques

2.4.1. Decision tree

A method of detecting depression by using a decision tree [22] is proposed. A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute. Each branch represents the outcome of the test, and each leaf node represents a class label. The paths from the root to the leaf represent classification rules.

The decision tree can be linearized into decision rules. The result is the contents of the leaf node, and the conditions along the path form a conjunction in the if clause. The rules have the form: if condition 1 and condition 2 and condition 3 then the outcome.

The AVEC 2014 dataset with depression measured by the BDI-II score is used in this study. Based on the works above, this study proposed a new method by using deep learning algorithms to analyze depression values with and without 3 affective dimension values.

2.4.2. K-nearest neighbor

The K-nearest neighbor (KNN) algorithm is one of the machine learning algorithms of classification [24]. It measures the Euclid distance between the test data and every training

data to decide the classification of the test data. This study adopts KNN to predict depression based on the AVEC 2014 training data.

An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors, details illustrated in Figure 2.8.

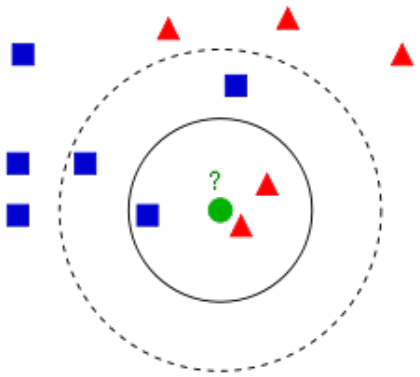


Figure 2.8: example of KNN [25]

In Figure 2.8, the green point needs to be classified into blue or red. In this case, k=3. The green dot finds its closest 3 neighbors. In the 3 neighbors, the majority number of the category is the green dot's category.

2.4.3. Neural networks

Neural networks have 3 types: multi-layer perception (MLP) neural network, convolutional neural network (CNN) and recurrent neural network (RNN). MLP neural network is used for classification and regression. The advantage of CNN is image classification and the advantage of RNN is time-series data analysis. This research uses MLP to calculate the 3-dimension affect values and CNN to analyze the discrete emotions and RNN to analyze depression during the whole HCI video.

The essential part of training a neural network is using the back-propagation algorithm to adjust weights and minimize the loss. Another training method is transfer learning which

is adding new layers based on a well-trained neural network and only trained the extended parts by using the back-propagation algorithm.

2.4.3.1. Multi-layer perception structure

Multi-layer perception (MLP) neural network is an interconnected group of nodes, shown in Figure 2.9, each node is a perception and an edge is a connection from the output of one proception to the input of another.

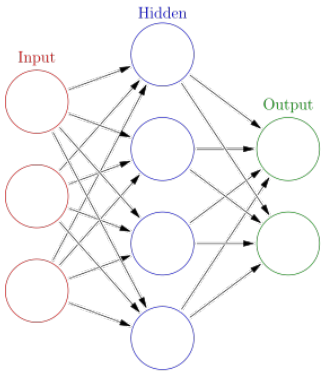


Figure 2.9: structure of MLP

In each node, a perceptron is an algorithm for supervised learning of binary classifiers. This algorithm enables neurons to learn and processes elements in the training set one at a time. Each perceptron includes a linear net input function and an activation function. The structure of a perceptron is shown in Figure 2.10.

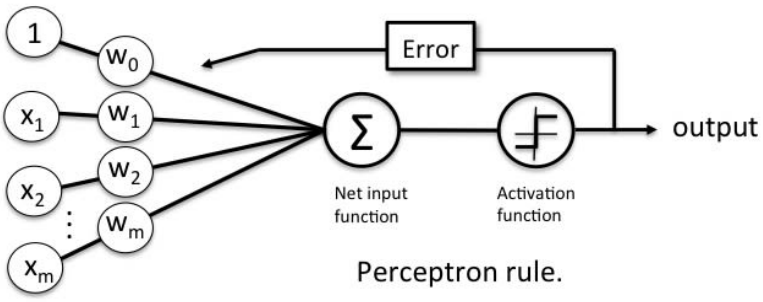


Figure 2.10: structure of perceptron [26]

The training data is a vector $[X_1,X_2,\ldots,X_m]$. Besides the input data, there is a bias value 1 feed into the perceptron. The net input function represents as:

$$Net\ input\ function = \sum_{i=1}^{m} x_i * w_i + w_0$$

There are some useful activation functions. In this study, ReLu, Softplus and SoftMax are used in different kinds of neural networks.

ReLu function is the Rectified Linear Unit function. If the input is x, then f(x)=max(0,x). The functional image is shown in Figure 2.11.

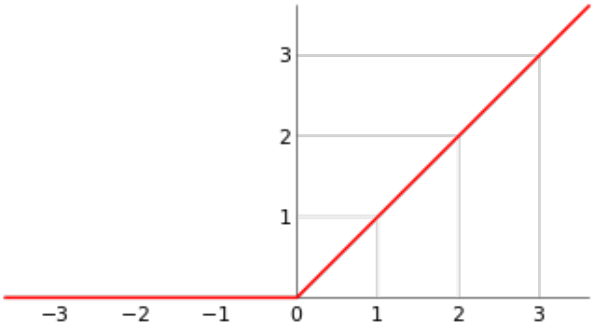


Figure 2.11: ReLu function

The softplus function, f(x)=ln(1+ex) is a smooth approximation to the Relu function. The functional image is illustrated in Figure 2.12.

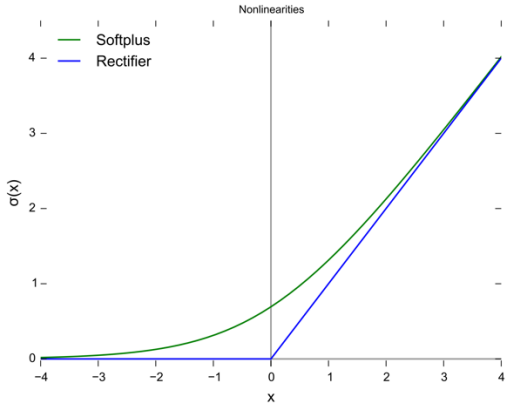


Figure 2.12: Softplus function

The softmax function, also known as normalized exponential function, is a function that takes as input a real number vector and normalized into a probability distribution. Before using softmax, some vector elements could be negative or greater than 1. The sum

might not to 1. After using softmax, each element of the vector will belong to the interval (0, 1). And the sum of all elements is 1. The softmax function is defined as:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

2.4.3.2. Convolutional neural network

Convolutional neural network (CNN) indicates that the network employs a mathematical operation called convolution, a specialized kind of linear operation. CNN uses convolution in place of general matrix multiplication in at least one of its layers.

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN usually consist of a series of convolutional layers. After a convolutional layer, the activation function is applied. The activation function is commonly a RELU layer. Then, pooling layers is applied to the activation function output. After multi convolutional layers, activation function and pooling layer. Some fully connected layers and normalization layers are added to the tail of the CNN.

A typical CNN called Lenet is designed to recognize the handwritten digit image [27]. There are five layers, including two convolutional layers, two fully connected layers and one output layer. The input size is 32x32; the output is a one-dimension vector with a length of 10. The structure of Lenet is shown in Figure 2.13.

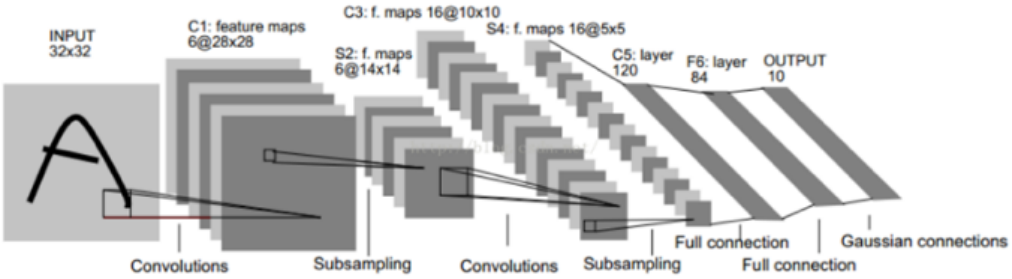


Figure 2.13: the structure of Lenet [27]

To apply CNN into a large scale image, the VGG19 CNN is proposed with input size 224x224 [17]. The image is calculated via multi convolutional layers with a 3x3 filter. One layer uses a 1x1 filter to represent a linear transformation of the input channel. The filter moving stride is 1 pixel. To preserved the size after the convolutional operation, padding with 1-pixel circumstance is added before convolution. Five max-pooling layers are carried out following some of the convolutional layers. The structure of VGG19 is illustrated in Figure 2.14

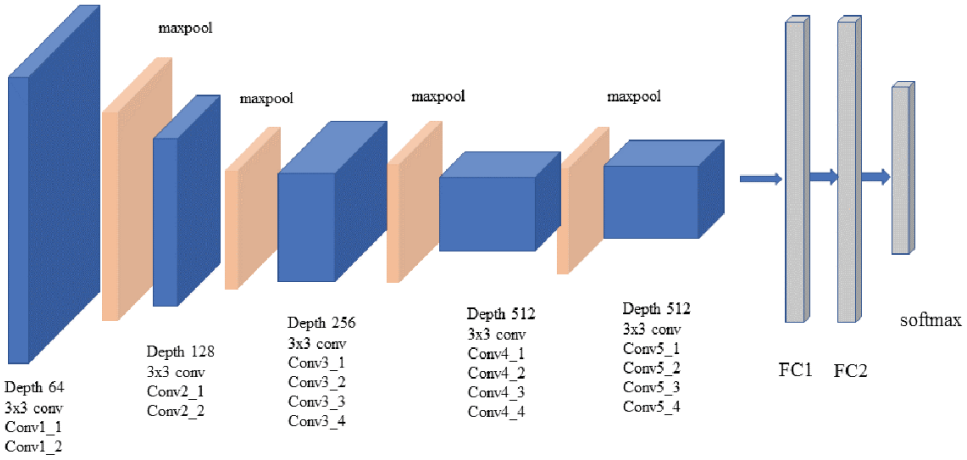


Figure 2.14: the structure of the VGG19 neural network [17]

2.4.3.3. Recurrent neural network

Recurrent neural network (RNN) is used for time series feature data analysis. RNN is consists of unfolding computational graphs which share parameters in the deep neural network structure. The basic RNN computational graph is shown in Figure 2.15.

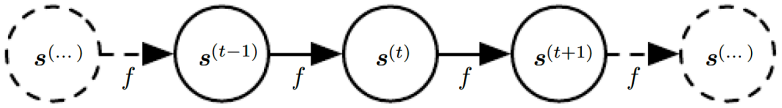


Figure 2.15: fundamental structure of RNN [28]

One node can be represented as:

$$s^{(t)} = f(s^{(t-1)}; \theta)$$

$S^{(t)}$ is the current state of the system.

Another structure is a system that drives an external signal $\mathbf{x}^{(t)}$. The current state is:

$$s^{(t)} = f(s^{(t-1)}, x^{(t)}; \theta)$$

The structure is illustrated in Figure 2.16.

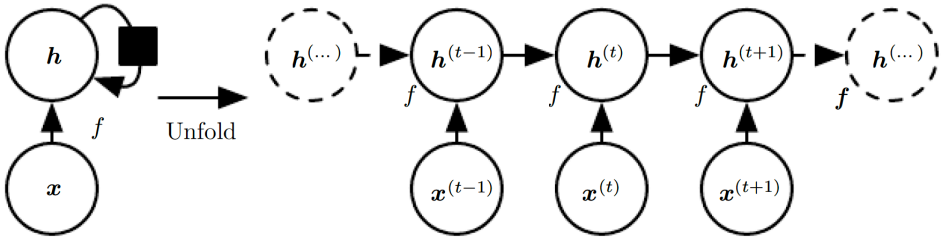


Figure 2.16: RNN with the external signals [28]

The left part of Figure 1.15 is the circuit structure. The black square means there is a 1-time step delay at the state t to state t+1. The unfolded graph is shown on the right side. Each time step is drawn a separate node in the computational graph. The size of the unfolded graph is fixed and depends on the input size.

The RNN is designed based on the structure in Figure 2.16. There are 3 different RNNs.

The first one is producing an output at each time step and has recurrent connection among hidden layers, illustrated in Figure 2.17.

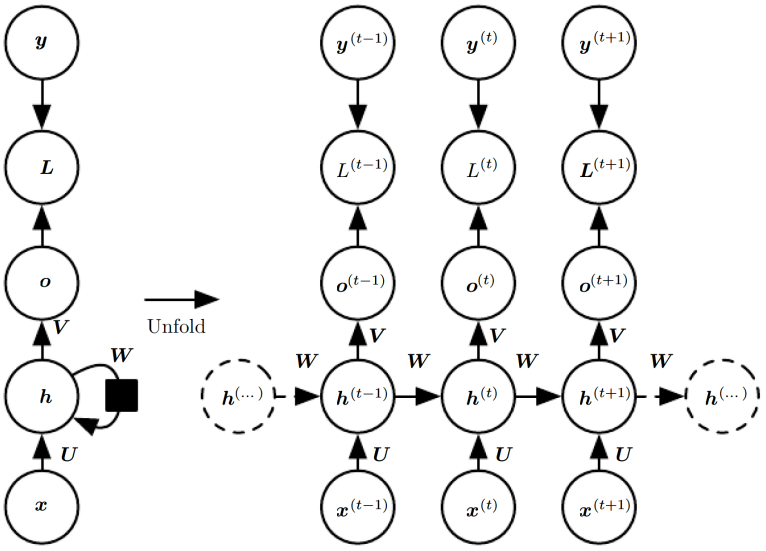


Figure 2.17: RNN produces an output at each time step [28]

Each node has an input vector $\mathbf{x}^{(t)}$ and the output vector $\mathbf{o}^{(t)}$. L is the loss function and $\mathbf{y}^{(t)}$ is the target output. There is a weight matrix **U** for each node input of the external signals. And the weight matrix **W** is used for calculating the previous hidden layer value. Weight matrix **V** is used for deciding the output value of each node.

The second one is producing an output at each time step and has connections from the output from the previous time step, shown in Figure 2.18.

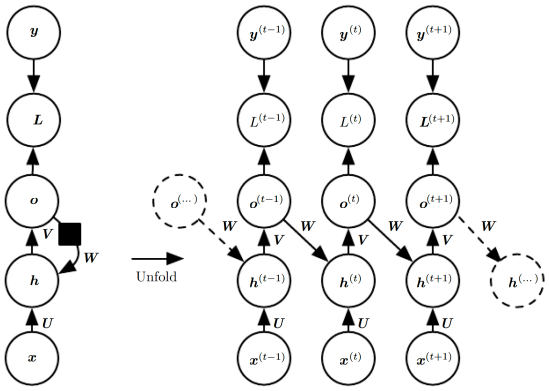


Figure 2.18: RNN only connection from the output of the previous layer [28]

This kind of RNN is worse than the first kind of RNN. The future units only depend on the output in previous units. The output usually loses information from past status.

Moreover, the hidden layer connection $\mathbf{h}^{(t)}$, represents the past input to the hidden layer and propagate to the future, cannot go forward.

The third one is only generating a single output. The structure is illustrated in Figure 2.19. In this study, the Long short-term memory (LSTM) neural network is designed based on this structure.

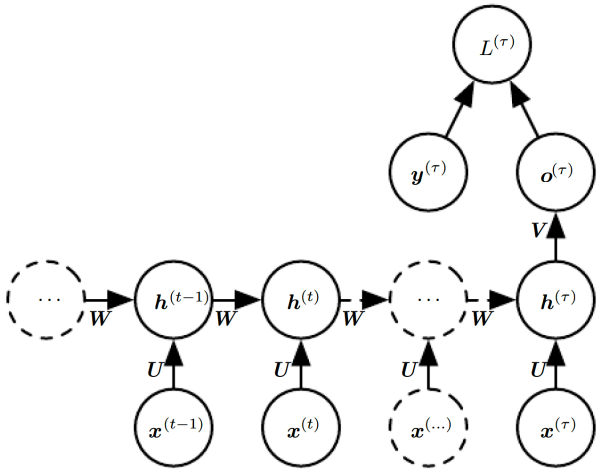


Figure 2.19: RNN only produce a single output [28]

In the LSTM neural network, the weight matric in each time can be changed instead of using the fixed matrix in Figure 1.18. Moreover, when previous information is used, it's better for the neural network to forget the old state by learning itself to decide when to do.

To allow neural networks to learn when to forget the previous, the forget is designed. The structure of the LSTM neural network is shown in Figure 2.20

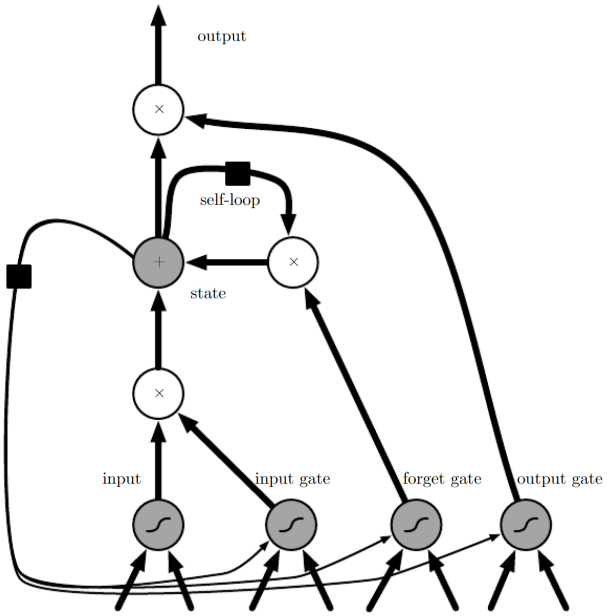


Figure 2.20: structure of LSTM neural network [28]

The state unit has a linear self-loop whose weight is controlled by the forget gate. All the gating units have a sigmoid active function. The black square indicates a delay of the 1-time unit.

The most important part of LSTM is the state unit $s_i^{(t)}$. The state unit has a liner self-loop which is calculated by a forget gate unit $f_i^{(t)}$ that its value is between 0 and 1 by using the sigmoid function.

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

The $\mathbf{x}^{(t)}$ is the current input vector and $\mathbf{h}^{(t)}$ is the current hidden layer vector. The $\mathbf{b}^f$, $\mathbf{U}^f$ and $\mathbf{W}^f$ are biases, input weights and recurrent weights of forget gates.

The internal states' weights are updated by using the formula below.

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right)$$

b, **U** and **W** are biases, input weights, and recurrent weights into the LSTM cell. $g_i^{(t)}$ is the input gate unit by calculated using the sigmoid function with parameters itself.

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

The output of the LSTM cell $h_i^{(t)}$ can be discarded by using the output gate $q_i^{(t)}$ with the sigmoid function.

$$h_i^{(t)} = \tanh\left(s_i^{(t)}\right) q_i^{(t)}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

2.4.3.4. Back-propagation algorithm

In Figure 2.10, there is an error computing based on the output. The error is defined as the difference between the perception output and the actual value. The difference is calculated by using loss function.

The most common loss is mean square error (MSE) loss, usually applied to regression tasks, which calculates the average e squared difference between the estimated values and the actual value. The MSE loss is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

For binary classification, the binary cross-entropy loss is better than MSE loss. the binary cross-entropy loss is defined as:

$$J(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right]$$

The propose of the back-propagation algorithm minimizes the error by changing weights in neural networks. Back-propagation computes the gradient of the loss function with respect to the weights of the network for a single input. The back-propagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule.

A detailed structure of a forward propagation process is illustrated in Figure 2.21.

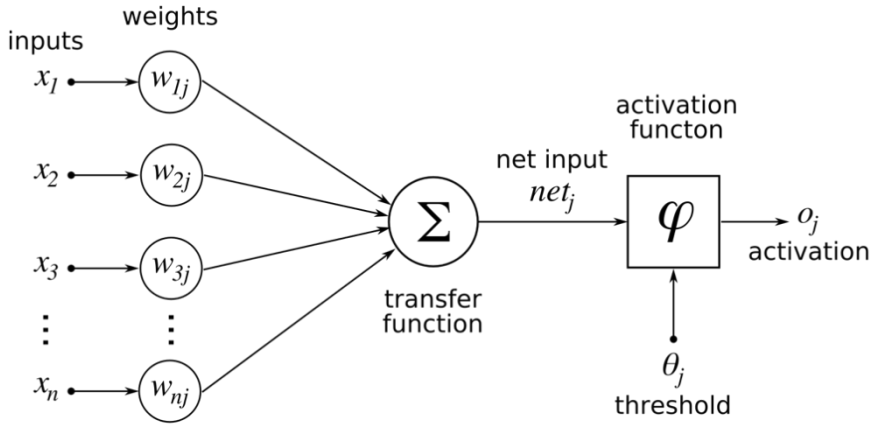


Figure 2.21: forward propagation process [26]

First, calculating the partial derivative of the error with respect to a weight w_{ij} by using the chain rule:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial \mathrm{net}_j}\frac{\partial \mathrm{net}_j}{\partial w_{ij}}$$

$$w_{new} = w_{old} - \eta\frac{\partial E}{\partial w_{ij}}$$

η is the learning rate, often in the range between 0 and 1.

2.4.3.5. Transfer learning

Transfer learning is a method for training neural networks by focusing on storing knowledge gained while solving one problem and applying it to a different but related problem. The input is the same but the output can be different. The transfer learning can

be achieved when there are features that are useful for different tasks corresponding to underlying factors that appear in more than one setting. The diagram of transfer learning is shown in Figure 2.22.

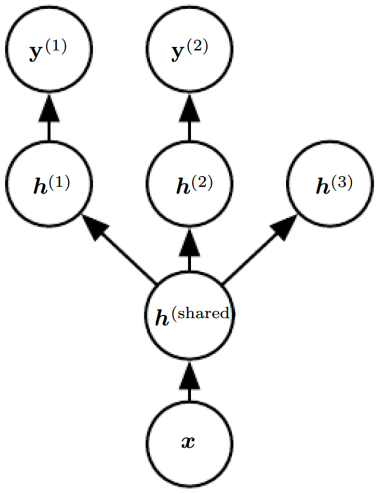


Figure 2.22: diagram of transfer learning [28]

x is the common input and $\mathbf{h}^{(\text{shared})}$ represents a trained neural network. $\mathbf{h}^{(i)}$ means some new layers extended to the trained neural network layers. $\mathbf{h}^{(i)}$ needs to be trained based on the network structure of $\mathbf{h}^{(\text{shared})}$ combining with $\mathbf{h}^{(i)}$. $\mathbf{y}^{(i)}$ is the output corresponding to $\mathbf{h}^{(i)}$.

In [29], the research proposed the transfer learning training data are independent and identity distributed with the test data. The advantage of transfer learning is it can solve problems with insufficient training data.

Transfer learning reuse the partial pre-trained neural network include the structure and pretrained weights. The first step is training a neural network based on the source domain dataset. The second step is keeping part of the pretrained structure and weights and connecting some new layers. The third step is training the new layers based on the target domain. The process is illustrated in Figure 2.23.

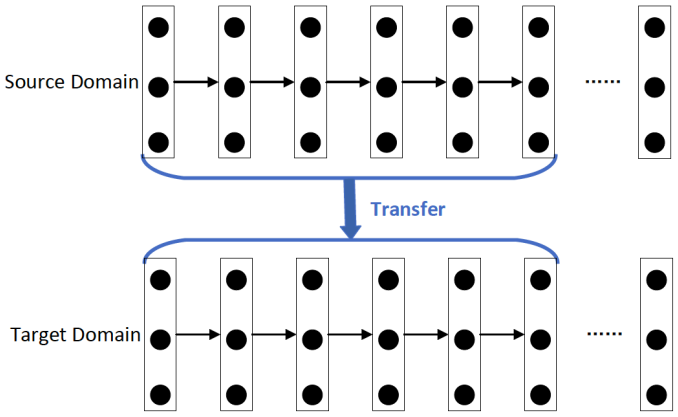


Figure 2.23: the process of transfer learning [29]

3. Human-computer interaction emotion analysis system

3.1. The sensor

The sensor of percepts the human face motion during the human-computer interaction is implemented by using OpenCV combine with the camera of Macbook pro. The OpenCV will open the camera and get the input from the camera and get frames of images at a constant frame rate. The steps are shown below:

Step 1. OpenCV request the operating system to access the camera

Step 2. OpenCV get input frames from the camera

Step 4. OpenCV call some functions to detect the location of face and analysis the emotion

3.2. The functions

The functions include 2 parts: facial location detection and facial expression recognition.

3.2.1. Facial location detection by using Dlib

Dlib provides facial location detection methods by impalement Max-Margin Object Detection (MMOD) [30] based on the histogram of oriented gradient (HOG) [31]. To detect the object in an image, a set of positive and negative image windows are selected from the training images. Then a binary classifier is trained on these windows. However, this method optimizes all sub-windows without performing any subsampling. The process is illustrated in Algorithm 1[31].

Algorithm 1 Object Detection

Input: image x, window scoring function f

$\mathcal{D}$:= all rectangles $r \in \mathcal{R}$ such that $f(x, r) > 0$
Sort $\mathcal{D}$ such that $\mathcal{D}_1 \geq \mathcal{D}_2 \geq \mathcal{D}_3 \geq ...$
$y^* := \{\}$
for $i = 1$ **to** $|\mathcal{D}|$ **do**
  if $\mathcal{D}_i$ does not overlap any rectangle in y^* **then**
    $y^* := y^* \cup \{\mathcal{D}_i\}$
  end if
end for
Return: y^*, The detected object positions.

r means a rectangular area of an image. R is the set of all rectangular areas scanned by the object detection system. To get the non-maximum suppression, the definition of no-overlap is defined as rectangles r_1 and r_2 do not overlap if the ratio of their intersection area to the total area covered is less than 0.5.

Y means the set of all valid labeling. Given an image x and a window scoring function f, e the object detection procedure can be defined as

$$y* = argmax_{y* \in Y} \sum_{r \in y} f(x, r).$$
$$f(x, r) = \langle w, \phi(x, r) \rangle$$

∅ is a HOG feature vector from the sliding window location r in the image x. w is a parameter vector.

The HOG uses linear SVM to do the binary classification. The process is illustrated in Figure 3.1.

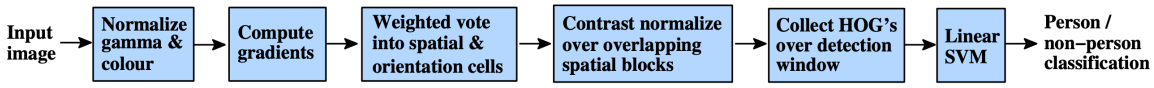


Figure 3.1: an overview of feature extraction and object detection [31]

The facial location detection is implemented by using the Dlib framework in python. To detect the face location in HCI videos, there exist some frames that the face cannot be detected by Dlib. Thus, based on the algorithm above, the location of the face in the previous frame is recorded. If the next frame cannot detect the face, then the sub-image in

the location of the previous frame recorded is used. In the beginning, Dlib will search all frames by the time order and start at the first image which can detect the face. The process is illustrated in Algorithm 2.

Algorithm 2 Face detection in HCI videos

Input: video frames
Step 1: Create a detector object by using Dlib framework
Step 2: Search images by using the detector in Step 1 and find the 1st frame that faces can be detected. Return the number of the frame
Step 3: Start at the frame in Step 2, detecting face location in the following frames.
Step 4: For each face in Step 3Analysis facial emotion

3.2.2. Facial expression recognition by using VGG19

After catching the face outline by using Dlib, a pre-trained VGG19 neural network is used to output one of the 7 emotions by using vector representation. The structure of the neural network is illustrated in Figure 1.12.

This study only needs the output of the final convolutional layer without the fully connected layer. Therefore, the fully connected layer was moved and the final convolutional layer was transformed into a length of 512 one dimension vector. The changed structure of the pre-trained VGG19 is shown in Figure 3.2.

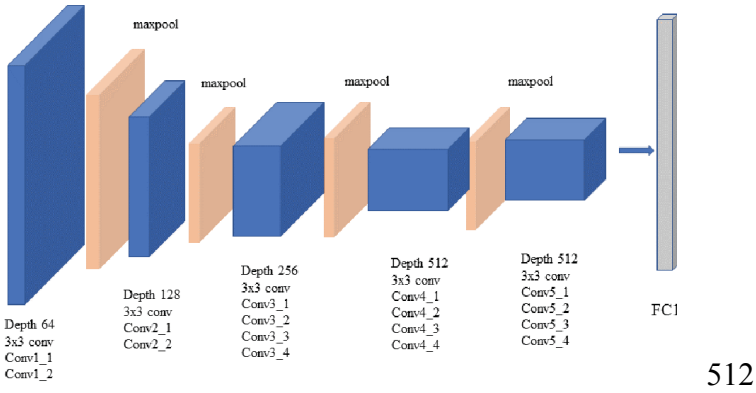


Figure 3.2: structure of the transformed VGG19 neural network

4. Affective dimensions analysis

4.1. Affective dimensions overview

The 2014 Audio-Visual Emotion Challenge dataset [19] is used to calculating continuous values of three affective dimensions (arousal, dominance and valence). The 3 affection dimensions can be represented as a vector of $[\vec{a}, \vec{d}, \vec{v}]$. The raw data of arousal, dominance and valence are belonging to the range of [-1000, 1000]. In this research, the data is scaled by a factor of 1/1000 to the range [-1,1].

There are 5 raters who marked all HCI data. The raters labeled the 3 dimension vector in continuous time and value using the FEELTRACE tool. Each video is rated by at least 3 people and at most 5 people.

To eliminate the difference between the 5 raters, the Inter-rater correlation coefficients (ICC) are computed combined with Pearson's r and RMSE. The One-vs-All inter-rater correlation coefficients, measured as Pearson's r and RMSE across all trace combinations is shown in Table 4.1 from [19].

Table 4.1: One-vs-All inter-rater correlation coefficients, measured as Pearson's r across all trace combinations [19]

X-vs-All	Arousal		Valence		Dominance		Average	
	r	RMSE	r	RMSE	r	RMSE	r	RMSE
A1	0.508	0.148	0.475	0.094	0.432	0.165	0.445	0.139
A2	0.474	0.159	0.624	0.083	0.422	0.178	0.319	0.146
A3	0.142	0.284	0.460	0.144	0.257	0.321	0.150	0.263
A4	0.505	0.190	0.627	0.150	0.400	0.211	0.474	0.186
A5	0.456	0.159	0.661	0.090	N/A	N/A	0.501	0.132

4.2. Affective dimensions prediction methods

An MLP neural network is implemented to learn the average trace based on all raters of each video. A new vector by connecting every 5 features is used for training to regress the label of 5th frame affective vector. The structure of the MLP neural network is shown in Figure 4.1

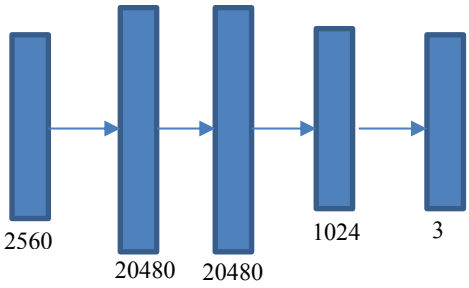


Figure 4.1: structure of the MLP for 3-dimension label prediction

The calculating processes are demonstrated below.

Step 1, get the feature vector from the middle layer from the Pre-trained deep convolutional neural network in each frame of the 2014 Audio-Visual Emotion Challenge video.

Step 2, a window of a vector of 2560 that combined 5 frames feature vectors is used for the neural network to predict the 3 affective dimension value of each 5th frame. Each window includes a 5x512 feature matrix and 3 affective dimension value of arose, violence and dominance.

Step3, after the first window is generated, the window moves forward by stride 1 to get another 5 frame feature vectors. The window continues to move forward until it reached the end frame.

The output values represent the value of the 3 dimensions. For each dimension, the average MSE loss of each training batch has been used to train the neural network. And the average MSE loss of each epoch of the test set is used to evaluate the effect of the whole neural network. The MSE loss is illustrated as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

yi represents the actual output of each value of the neural network and $\hat{y}_i$ represents the label of each value of the 3 dimensions. N is the batch size.

4.3. Discussion

There are 49 videos from the AVEC 2014 dataset. After generating the 5x512 vector for training, there are around 80000 datasets includes 75000 for training and 5000 for testing. The benchmark of the MSE loss of Arousal, Dominance, and Valence are illustrated in Table 4.2 from [19]. The average loss for training and testing after each epoch is shown in Table 4.3.

Table 4.2: One-vs-All inter-rater correlation coefficients, measured as Pearson's r across all trace combinations [19]

X-vs-All	Arousal		Dominance		Valence	
	r	MSE	r	MSE	r	MSE
A1	0.508	0.021904	0.432	0.027225	0.475	0.008836
A2	0.474	0.025281	0.422	0.031684	0.624	0.006889
A3	0.142	0.080656	0.257	0.103041	0.460	0.020736
A4	0.505	0.0361	0.400	0.044521	0.627	0.0225
A5	0.456	0.025281	N/A	N/A	0.661	0.0081
Average		0.035		0.048		0.013

Table 4.3: average loss for training and testing after each epoch (our work)

Epoch	Train Arousal	Train Dominance	Train Valence	Test Arousal	Test Dominance	Test Valence
1	0.007962319	0.012923535	0.012004063	0.02843411	0.020956658	0.02083081
2	0.005430296	0.007861455	0.007835883	0.034782376	0.024797672	0.023746962
3	0.004647697	0.006728826	0.006711911	0.029743109	0.023130532	0.02446534
4	0.004171305	0.005936381	0.005921872	0.03332632	0.019395962	0.01969264
5	0.00376158	0.00524994	0.005241691	0.033324543	0.019785034	0.02034393
6	0.003449357	0.004698326	0.004691762	0.034795646	0.020153929	0.019749233
7	0.003176497	0.00425084	0.004242935	0.03481131	0.022656389	0.023131588
8	0.002914457	0.003816382	0.003810396	0.034814708	0.022909982	0.023022242
9	0.00268708	0.003454648	0.003448283	0.03264354	0.024445172	0.024187768
10	0.002483581	0.003085688	0.003076027	0.03302835	0.024079153	0.024012139

The result shows 1) in the 10 epochs training process, the training MSE loss of the 3 dimension value decreases; 2) the testing MSE loss keeps steady; 3) after 10 epochs training, the test MSE loss for Arousal, Dominance and Valence are 0.03302835 0.024079153 0.024012139, which are close to the benchmark of the average MSE loss.

5. Depression analysis by using affective dimensions

5.1. Affective dimensions data analysis

In the 2014 Audio-Visual Emotion Challenge dataset, there are 4 types of degree of depression which are no or minimal depression, mild depression, moderate depress and severe depression. The depression value for each person of the interaction video is obtained by the Beck Depression Inventory-II (BDI-II) questionnaire score [32]. The final BDI-II scores range from 0–63. Ranges can be interpreted as follows: 0–13 (D0): indicates no or minimal depression, 14–19 (D1): indicates mild depression, 20–28 (D2): indicates moderate depression, 29–63 (D3): indicates severe depression.

Each video was assigned a depression degree according to the total interaction process. The pilot study analyzed the relationship between the 3-dimension affect label and the depression label by visualizing the 3-dimension affect labels as a coordinate in the Euclidian space. Figure 5.1 visualizes all 3-dimension affect coordinates of each video. Figure 5.2 visualizes the average 3-dimension affect the coordinate of each video. Each color represents a type of depression. X, Y, Z corresponds to the dimension of arousal, dominance and valence.

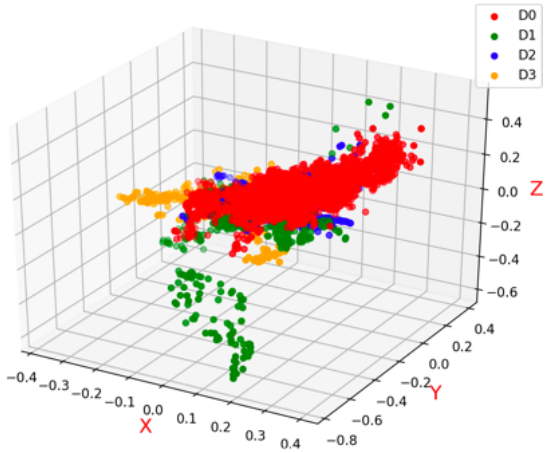


Figure 5.1: visualize all vertices of each video

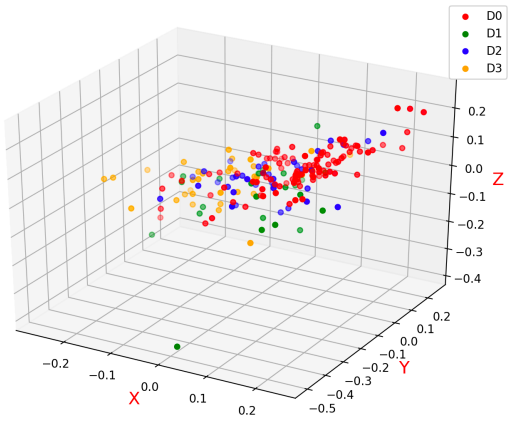


Figure 5.2: visualize average value vertex of each video

The figures above illustrated that: 1) all 3-dimension affective vertices are located in a specific plain; 2) there is no significant difference between the vertices of the 4 depression type.

Due to the depression label is related to the all 3-affection dimension within a time series, the LSTM neural network is proposed to train the depression degree from the 3 dimension affective label obtained from the real human-computer interaction time series.

5.2. Depression degree prediction based on affective label

The depression degree is predicted from the 3-demission affective labels during a constant time series by using Long short-term memory (LSTM) neural network. The first step is to change all the time period affective labels into a constant length time series sequence. The constant length is defined by the maximum time period video. For the time less than the maximum time, enough zeros will be added into the front of each affective label sequence. The second step is creating an LSTM neural network to train the affective label from the input of the constant length of affective labels time series and output the depression value based on regression. Finally, the performance of the LSTM will be evaluated and the alternative method is proposed to improve the depression degree analysis effect.

First, modify each data into the same size of 7440x3, that is, the maximum of the number of frames in the dataset times the 3-dimension affective label. A sample of training data is illustrated in Table 5.1.

Table 5.1: sample of training data

Arousal	Dominance	Valence
0.0972	0.06825	0.0024
0.1004	0.06825	0.0076
0.102	0.06825	0.0116
0.1084	0.06825	0.0154
0.11107	0.06825	0.020734
0.1164	0.06825	0.0285

The training data will be inputted into LSTM to regress the output of depression value. LSTM is a recurrent neural network structure in deep learning. Compare to traditional feedforward neural networks, LSTM has feedback connections. It can process single data points, but also entire sequences of data. There are several architectures of LSTM units. A common architecture is composed of a cell (the memory part of the LSTM unit) and three "regulators", usually called gates, of the flow of information inside the LSTM unit: an input gate, an output gate and a forget gate. The basic structure of the LSTM is shown in Figure 5.3.

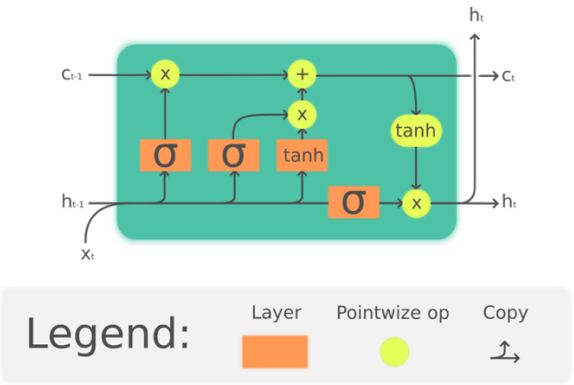


Figure 5.3: structure of LSTM

In this research, the structure of the 70 LSTM unit network is implemented. The output is a float number represented the depression value.

5.3. Discussion

There total 300 number of data, includes 200 for training, 50 for validation and 50 for testing. There are a total of 2000 training epochs includes a batch size of 30 data in each epoch. The learning rate is $0.1x10^{-3}$ with decay $0.1x10^{-3}$ at each training epoch. The mean square error (MSE) loss is used for backpropagation.

At the beginning of the training, the validation loss decreases dramatically. After 100 training epochs, the validation loss represents periodicity vibrate from 100 to 110. The validation loss is illustrated in Figure 5.4.

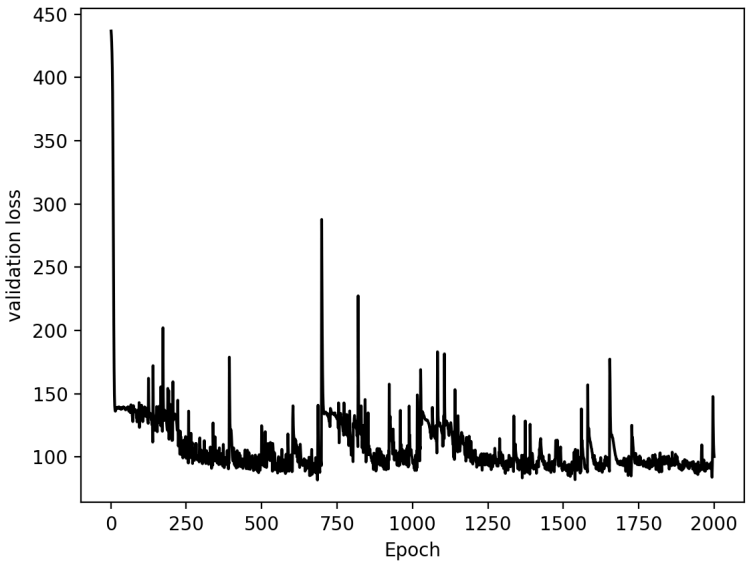


Figure 5.4: validation loss during training

After 2000 epoch training, the best epoch of the minimum validation loss (MSE 81.79 at the 685th epoch) is used for testing. For binary classification, the accuracy is 58% and the 4-type classification is 50%, respectively. The confusion matrix for the test dataset is illustrated in Table 5.2 and Table 5.3.

Table 5.2: confusion matrix of binary classification

	Predict: 0	Predict: 1
Actual: 0 (Total: 32)	23	9
Actual: 1 (Total: 18)	12	6

Table 5.3: confusion matrix of 4-type classification

	Predict: 0	Predict: 1	Predict: 2	Predict: 3
Actual: 0 (Total: 29)	23	6	3	0
Actual: 1 (Total: 10)	7	2	1	0
Actual: 2 (Total: 8)	4	0	0	0
Actual: 3 (Total: 7)	1	2	1	0

In Table 5.2, label 0 represents no depression and 1 means depression. In Table 5.3, label 0 represents no depression and 1-3 correspond to low, middle and high depression. And the number represents the number of prediction data that match the actual label.

In the testing dataset, the result illustrated this algorithm is a promising method. In the row of actual label 0 from Table 5.2, the recall rate for the non-depression type is 71% and for depression type is 33% which illustrated the neural network can distinct depression users and non-depression users.

To improve recognition accuracy, a new method of analysis users' depression based on the head and facial muscle movement instead of the psychological 3-dimension affective is proposed in Chapter 6.

6. Depression analysis from HCI videos

In this section, the traditional LSTM neural network is used to predict depression. The LSTM structure is similar to the structure in chapter 5. Only the difference is the input is a time series with a length of 512 feature vectors instead of 3 affective dimension labels. Moreover, an improved method based on the relationship between negative emotions and depression is proposed. Another neural network is implemented for the prediction of depression and non-depression.

6.1. The relationship between depression and facial expressions

In this section, the pattern of emotions during the human-computer interaction process is analyzed. Emotions are defined as positive emotions Happy, Surprise and Neutral) and negative emotions (Angry, Disgust, Fear and Sad). In the HCI video time series, positive emotions are noted as 0 and negative emotions are marked as 1. The proportion of each emotion is used to distinguish depression and without depression.

6.1.1. The relationship between the proportion of emotions and depression

In this section, the first step is calculating the percentage of each emotion in a video. Second, for each level of depression, the average percentage of each emotion is calculated as a feature vector of the depression level. Third, the Euclid distance matrix which represents the distance from one depression level to another. The distance matrix for 4 type depression is shown in Table 6.1. There is a total of 198 samples because there are 2 samples that only detect neutral emotion but the bias between depression is very large. The number of each type of depression is shown in the pretenses.

Table 6.1: distance matrix for 4 type depression

	0 (104)	1 (24)	2 (36)	3 (34)
0	0	0.11	0.14	0.18
1	0.11	0	0.10	0.11
2	0.14	0.10	0	0.10
3	0.18	0.11	0.10	0

In the 198 samples, there are 104 samples have no depression and 94 have depression. The distance between depression and no depression is 0.13. Table 6.1 illustrates it's can recognize into 2 classes.

6.1.2. Algorithm design

In this chapter, the neural network and K-nearest neighbor (KNN) are considered for classification. In the neural network method, the structure is LSTM and the input is a sequence of the length of 512 vector that represents the emotion. Regression and classification training is implemented to find the best model. The KNN methods are calculated the nearest distance to the 4 depression feature vector.

6.2. Neural network methods

In this section, there are 2 methods in depression analysis by using facial features: regression and classification. The facial feature is a length of 512 vector received from the transformed VGG19 neural network in Figure 3.2. The regression method tested depression prediction into 4 categories and 2 categories. And classification method tested the performance of classification into 2 categories.

6.2.1. Regression

A regression LSTM neural network with an input raw length of 512 vector introduced in Chapter 2 is implemented. A length of 512 of each video frame can be calculated from the VGG19 in Chapter 2. The preprocessing is introduced in Chapter 4. In the 200 epochs

training process, the loss function is MSE loss and the optimizer is Adam optimizer with learning rate 0.001. The loss of validation dataset during the training process is shown in Figure 6.1.

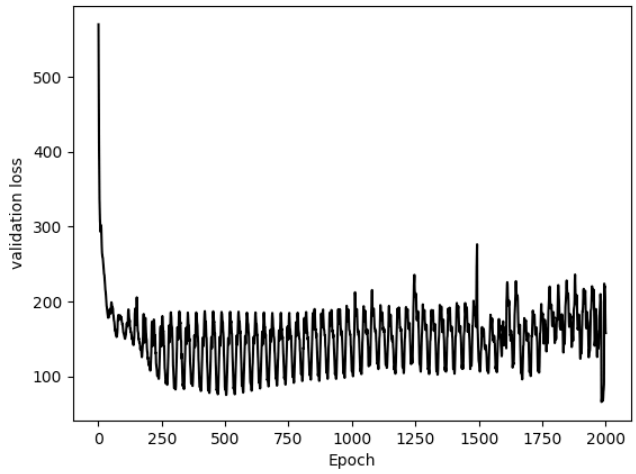


Figure 6.1: validation loss during training

During the training process, the validation loss keeps around 100 to 200. After 2000 training epochs, the minimum validation loss located at the 1983 epoch is 66.1.

For binary classification, the accuracy is 56% and the 4-type classification is 42%, respectively. The confusion matrix for the test dataset is illustrated in Table 6.2 and Table *6.3*.

Table 6.2 confusion matrix of binary classification

	Predict: 0	Predict: 1
Actual: 0 (Total: 32)	15	17
Actual: 1 (Total: 18)	5	13

Table 6.3: confusion matrix of 4-type classification

	Predict: 0	Predict: 1	Predict: 2	Predict: 3
Actual: 0 (Total: 29)	15	12	5	0
Actual: 1 (Total: 10)	2	5	3	0
Actual: 2 (Total: 8)	2	1	1	0
Actual: 3 (Total: 7)	1	2	1	0

6.2.2. Classification

The same structure of the LSTM neural network is used for 2 classes of depression prediction. The active function of the output layer is SoftMax and the loss function is binary cross-entropy loss. There are 2000 training epochs with batch size 30, learning rate 0.001 and Adam optimizer. The validation loss and accuracy trends are shown in Figure 6.2: validation loss during trainingFigure 6.2 and Figure *6.3*.

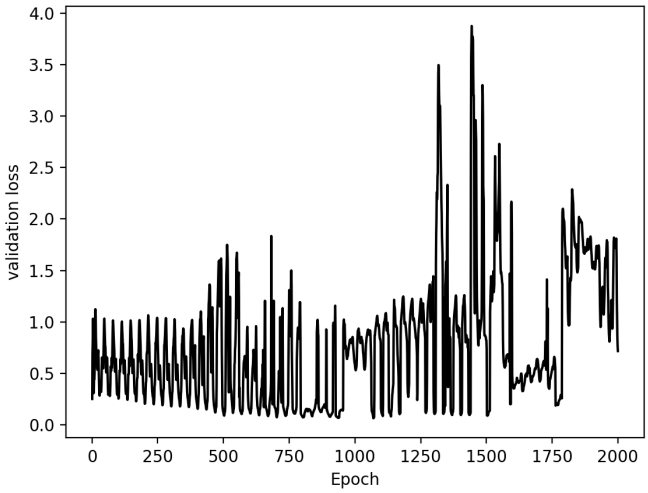


Figure 6.2: validation loss during training

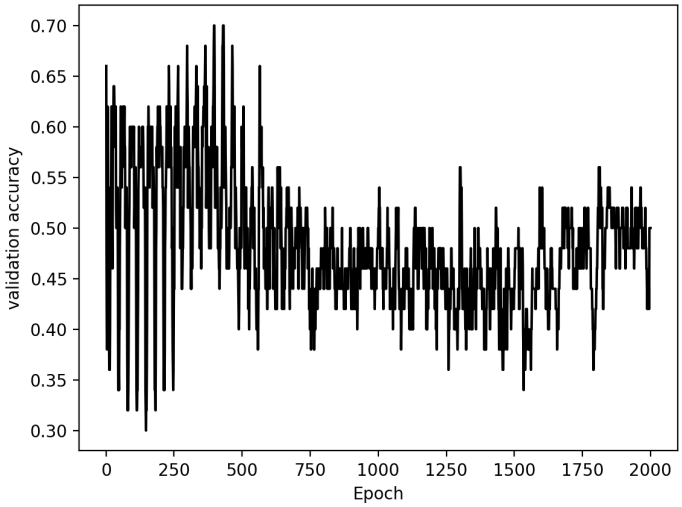


Figure 6.3: validation loss during training

The best epoch is the 397th epoch with a validation accuracy of 70%. For binary classification on the testing dataset, the accuracy is 46%. The confusion matrix is shown in Table 6.4.

Table 6.4: confusion matrix of binary classification

	Predict: 0	Predict: 1
Actual: 0 (Total: 32)	11	21
Actual: 1 (Total: 18)	6	12

6.3. KNN method based on the emotion analysis

Based on the four feature vectors of the depressions, the K-nearest neighbor (KNN) algorithm is proposed for classification. The training dataset is the set that includes the 198 samples, introduced in section 5.1.1. The validation and testing set, including 50 samples for each dataset, are the same as neural network evaluation tasks. In the beginning, the continuous values of depressions are transformed into the category value 0, 1, 2, 3. The vector of the percentage of each emotion and the category depression value is inputted to the KNN classifier. This study proposed KNN based on the individual data and based on the average feature vector of 4 types of depressions.

6.3.1. KNN based on the individual data

The validation dataset is used for finding the best value of k. There are 50 validation and test samples, so finding the best k value is evaluating the validation dataset by choosing k from each integer from [1, 50]. The accuracy is shown in Figure 6.4.

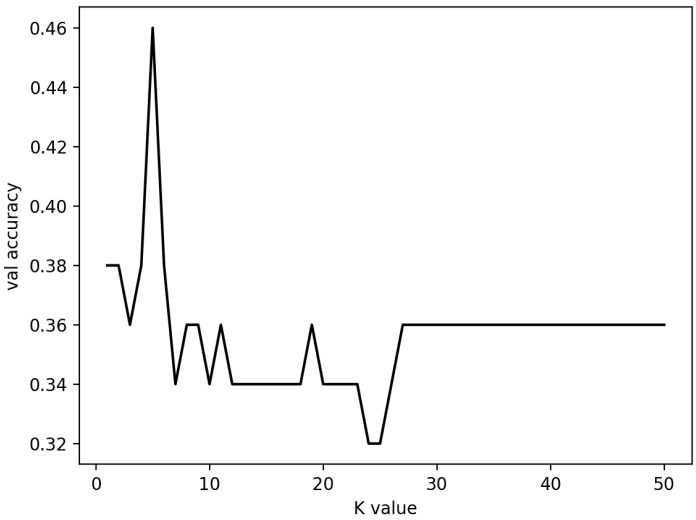


Figure 6.4: validation loss by choosing different k value

Figure 6.4 illustrates the best k is 5 and the validation accuracy is 46%. From k=6 to 26, the accuracy shows fluctuation. After k=27, the accuracy is fixed to 36%.

In this study, k is tested by k=5 and k=27. When k=5, the test dataset accuracy is 62% and when k=27, the test dataset accuracy is 70%. The confusion matrices are shown in Table 6.5 and Table 6.6.

Table 6.5: confusion matrix at k=5, accuracy=62%

	Predict: 0	Predict: 1
Actual: 0 (Total: 32)	27	5
Actual: 1 (Total: 18)	14	4

Table 6.6: confusion matrix at k=27, accuracy=70%

	Predict: 0	Predict: 1
Actual: 0 (Total: 32)	31	1
Actual: 1 (Total: 18)	14	4

From the 2 tables above, k=27 is better than k=5. When k=27, the classification of non-depression has improved.

6.3.2. KNN based on the standard feature vectors

For each type of depression, the feature vector is calculated by the average vector of the percentage of each emotion. Therefore, there are 4 standard feature vectors represent each type of emotion. In each sample in the dataset, the closest Euclid distance feature vector's category is its category. The accuracy is 50% and the confusion matrix is shown in Table 6.7.

Table 6.7: confusion matrix for KNN on standard feature vectors, accuracy=50%

	Predict: 0	Predict: 1
Actual: 0 (Total: 32)	12	20
Actual: 1 (Total: 18)	5	13

The accuracy is less than the methods by using individual data. For the effects in classification, the recall of depression is better than the methods by using individual data.

6.4. Discussion

In this chapter, this study analysis emotion features in the HCI video and proposed neural network and KNN methods for binary classification. In neural network methods, regression training based on the continuous depression label and classification training by using the transformed 4 type depression is implemented. In KNN methods, this study tests the classification based on each sample and the classification based on the standard feature vectors for the 4-type depression. All experiments demonstrate the binary classification accuracy is higher than the 4-type classification.

During the neural network training process, the loss for the validation dataset shows serves to vibrate which means the neural network doesn't have a perfect learning effect. Comparing the regression method and the classification method, regression accuracy 56%

is higher than the classification accuracy of 46%. However, in the confusion matrices in Table 6.2 and

Table *6.3*, the classification methods demonstrate the advantage in the classification of non-depression type. Therefore, the neural network trained by classification is better than regression.

In KNN classification, this study proposed 2 classification methods, one is finding the best k value based on each training sample and the other is finding the closest Euclid distance feature vector.

In the first method, the best k value is 27 which means the data will find the closest 27 neighbors and find the majority category. The accuracy is 70%. This method illustrates the advantage of recognizing non-depression but and disadvantage to recognize depression video.

The second method shows a good performance in recognizing the depression value. However, there is more error in distinguishing non-depression value. The accuracy of 50% means this method demonstrates a random classification.

Comparing to the 4 methods of using neural network and KNN, the best performing method is KNN binary classification when k=27.

7. Conclusions and future works

7.1. Models evaluation

This study proposed 2 directions of detection depression based on HCI videos. The first is analyzing depression based on the 3-dimensions affective values (Arousal, Dominance and Valence). The second one is analyzing depression based on the discrete emotion features from each video frame.

In the first track of using 3-dimension value, the first step is constructing an MLP neural network to generate the 3-dimension value for each HCI video for each frame. The MSE loss is calculated and compared to the benchmark. The result illustrates the algorithm of calculating the 3-dimension value is an effective method.

The second step is predicting the depression value based on each frame's 3-dimension values proposed in the AVEC 2014 dataset instead of values generating by ourselves. However, the accuracy of 58% shows it cannot accurately distinguish depression and non-depression. Moreover, there are more error once the 3-dimension values generated automatically is used for predicting the depression. Furthermore, 3-dimension values in AVEC 2014 dataset are evaluated by human so that it demonstrates some subjectivity and cannot accurately represent the emotions and depressions.

The second track is analyzing depression by extracting discrete emotion features in HCI videos by using 2 methods. The first one is extracting the emotion vector generated by the VGG19 emotion recognition neural network. Then LSTM neural network is implemented for depression prediction. In the training process, 2 methods (regression and classification) are proposed. The result shows the classification method has better performance than

regression methods. However, the accuracy is below 50% so that the LSTM classification cannot build into real clinic experiments.

The second one is defining a depression feature vector for each HCI video. Each element in the vector is the percentage of a specific emotion from the 7 categories. KNN algorithm is implemented for classification. There are 2 implementations. The first one is the standard KNN algorithm based on each labeled training data, finding the best k value of the number of closest neighbors needed to be considered and counting the majority number of categories within those neighbors. The second one is building a standard feature vector for each level of depression. The Euclid distance from the test data to each of the standard feature vector is calculated and finding the closest distance standard feature vector as its category of depression. The first method experiment shows the best k value is 27 and the accuracy is 70% which is better than the second one of 50% by using standard feature vectors.

The best model in this study is the standard KNN classification on k=27 based on the 198 training samples. A baseline from AVEC 2016 [33] dataset is shown in Table 7.1.

Table 7.1: AVEC 2016 baseline [33]

Partition	Modality	F1 score	Precision	Recall
Development	Audio	.462 (.682)	.316 (.938)	.857 (0.54)
Development	Video	.500 (.896)	.600 (.867)	.428 (.928)
Development	Ensemble	.500 (.896)	.600 (.867)	.428 (.928)
Test	Audio	.410 (.582)	.267 (.941)	.889 (.421)
Test	Video	.583 (.851)	.467 (.938)	.778 (.790)
Test	Ensemble	.583 (.857)	.467 (.938)	.778 (.790)

In Table 7.1, the recall for depression and non-depression are reported. Values for non-depression are reported in the bracket. The average recall baseline of development and test video dataset for none depression is 85.9% and the baseline for depression is 60.3%. The

recall for none depression in our proposed KNN algorithm is 96.8% which is higher than the baseline.

In our proposed KNN method, the recall of 96.8% and the accuracy 70% demonstrates this is a reliable method can be used in actual environments.

7.2. Conclusion

This study has developed and implemented a human-computer interaction depression detection system by using machine learning techniques. To predict the depression degree, the first step is using a pre-trained VGG19 neural network to analyze the user's emotion in each video frame from AVEC 2014 dataset. The second step is transforming the continuous depression value to discrete 4 types of depressions and trained a k nearest neighbor classifier based on 198 labeled training data from AVEC 2014 dataset. The result shows that comparing to the 27 nearest neighbors is the best method with accuracy 70%. With further enhancement, our method can be used for real-world applications.

7.3. Future works

This model has some disadvantages because it cannot precisely detect users that have the trend of depression. In the current research, only video frame data are used. In the HCI process, there are other information can be used for depression analysis. To improve the depression detection accuracy, more information on video frames, audio features and text features could be considered.

A lot of potential information on the video frame can be extracted. Research [22] shows there are different rules based on gender. In our future plan, males and females could be considered to train separately on different machine learning classifiers. Another important feature is details on facial movement. Facial landmarks, eye gaze features and head pose

features are helpful on emotion recognition. We could improve the quality of training data as follows. The first step is calculating the mean shape of some stable points that can describe facial muscle movement. Then, the feature points can be aligned with the mean shape. Finally, the Euclid distance and the angle between the mean shape points and aligned face points can be considered a useful feature for training.

Audio features perform an important rule in depression decision making [34]. Psychologists use audio information in the diagnosis of depression. Moreover, the speech signals are convenient for sampling and training.

Besides the video and audio features, texts of audios are an important factor to diagnose depression. The text sudden abbreviations are an important feature of negative emotions. The semantic of texts can also be used to diagnose depression.

8. Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisor Professor Dr. Liang Liang for the continuous support of my master's study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis, especially have me enormous suggestions about technology conceives of machine learning. I could not have imagined having a better supervisor and mentor for my master's study.

Besides the supervising of this research, I would like to appreciate that Professor Dr. Liang taught the course of machine learning. It gave me so much knowledge and techniques for this research. Moreover, I would like to thank Professor Dr. Ubbo Visser taught me the course of the introduction of artificial intelligence. From this course, I came up with the idea of this research based on the chapter of the robot and agent system.

I also sincere thanks to Professor Dr. Vissor. As the director of graduate studies, he patiently gave me advice on this research and agreed on my research course registration. He also gave me precious suggestions on the plan of this research. Without Professor Dr. Vissor’s long-term support, I could not make great achievements.

9. REFERENCES

[1] Health & Consumer Protection Directorate General, "GREEN PAPER- Improving the mental health of the population Towards a strategy on mental health for the European Union," p. 30.

[2] European Commission and Statistical Office of the European Union, *Labour market policy expenditure and participants: data 2006.* Luxembourg: Publications Office, 2008.

[3] T. M. Press, "Affective Computing | The MIT Press." https://mitpress.mit.edu/books/affective-computing (accessed May 09, 2020).

[4] H. Abdollahi, A. Mollahosseini, J. T. Lane, and M. H. Mahoor, "A Pilot Study on Using an Intelligent Life-like Robot as a Companion for Elderly Individuals with Dementia and Depression," *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 541–546, Nov. 2017, doi: 10.1109/HUMANOIDS.2017.8246925.

[5] S. Al Moubayed, J. Beskow, G. Skantze, and B. Granström, "Furhat: A Back-Projected Human-Like Robot Head for Multiparty Human-Machine Interaction," in *Cognitive Behavioural Systems*, Berlin, Heidelberg, 2012, pp. 114–130, doi: 10.1007/978-3-642-34584-5_9.

[6] R. Simmons *et al.*, "GRACE: An Autonomous Robot for the AAAI Robot Challenge," *AI Magazine*, vol. 24, no. 2, pp. 51–51, Jun. 2003, doi: 10.1609/aimag.v24i2.1704.

[7] D. C. Herath, C. Kroos, C. J. Stevens, L. Cavedon, and P. Premaratne, "Thinking head: Towards human centred robotics," in *2010 11th International Conference on Control Automation Robotics & Vision*, Singapore, Singapore, Dec. 2010, pp. 2042–2047, doi: 10.1109/ICARCV.2010.5707899.

[8] S. J. Russell, P. Norvig, and E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.

[9] M. Valstar *et al.*, "AVEC 2013: the continuous audio/visual emotion and depression recognition challenge," in *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge - AVEC '13*, Barcelona, Spain, 2013, pp. 3–10, doi: 10.1145/2512530.2512533.

[10] C. Lisetti, R. Amini, and U. Yasavur, "Now All Together: Overview of Virtual Health Assistants Emulating Face-to-Face Health Interview Experience," *Künstl Intell*, vol. 29, no. 2, pp. 161–172, Jun. 2015, doi: 10.1007/s13218-015-0357-0.

[11] P. Peña, M. Polceanu, C. Lisetti, and U. Visser, "eEVA as a Real-Time Multimodal Agent Human-Robot Interface," in *RoboCup 2018: Robot World Cup XXII*, vol. 11374, D. Holz, K. Genter, M. Saad, and O. von Stryk, Eds. Cham: Springer International Publishing, 2019, pp. 262–274.

[12] W. Liu, C. Song, and Y. Wang, "Facial expression analysis using a sparse representation based space model," in *2012 IEEE 11th International Conference on Signal Processing*, Oct. 2012, vol. 3, pp. 1659–1662, doi: 10.1109/ICoSP.2012.6491899.

[13] H. P. Mal and P. Swarnalatha, "Facial expression detection using facial expression model," in *2017 International Conference on Energy, Communication, Data*

Analytics and Soft Computing (ICECDS), Aug. 2017, pp. 1259–1262, doi: 10.1109/ICECDS.2017.8389644.

[14] S. Agarwal and D. P. Mukherjee, "Decoding mixed emotions from expression map of face images," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Apr. 2013, pp. 1–6, doi: 10.1109/FG.2013.6553731.

[15] B. Fasel, "Robust face analysis using convolutional neural networks," in *Object recognition supported by user interaction for service robots*, Aug. 2002, vol. 2, pp. 40–43 vol.2, doi: 10.1109/ICPR.2002.1048231.

[16] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Apr. 1998, pp. 200–205, doi: 10.1109/AFGR.1998.670949.

[17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Sep. 2014, Accessed: Mar. 18, 2019. [Online]. Available: http://arxiv.org/abs/1409.1556.

[18] H. Jun, L. Shuai, S. Jinming, L. Yue, W. Jingwei, and J. Peng, "Facial Expression Recognition Based on VGGNet Convolutional Neural Network," in *2018 Chinese Automation Congress (CAC)*, Nov. 2018, pp. 4146–4151, doi: 10.1109/CAC.2018.8623238.

[19] M. Valstar *et al.*, "AVEC 2014: 3D Dimensional Affect and Depression Recognition Challenge," in *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, New York, NY, USA, 2014, pp. 3–10, doi: 10.1145/2661806.2661807.

[20] R. Cowie, E. Douglas-Cowie, S. Savvidou, E. M. M. Sawey, and M. Schroeder, "'FEELTRACE': AN INSTRUMENT FOR RECORDING PERCEIVED EMOTION IN REAL TIME," p. 7.

[21] E. W. Martinsen, S. Friis, and A. Hoffart, "Assessment of depression: comparison between Beck Depression Inventory and subscales of Comprehensive Psychopathological Rating Scale," *Acta Psychiatr Scand*, vol. 92, no. 6, pp. 460–463, Dec. 1995, doi: 10.1111/j.1600-0447.1995.tb09613.x.

[22] L. Yang, D. Jiang, L. He, E. Pei, M. C. Oveneke, and H. Sahli, "Decision Tree Based Depression Classification from Audio Video and Language Information," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge - AVEC '16*, Amsterdam, The Netherlands, 2016, pp. 89–96, doi: 10.1145/2988257.2988269.

[23] G. Stratou, S. Scherer, J. Gratch, and L.-P. Morency, "Automatic nonverbal behavior indicators of depression and PTSD: the effect of gender," *J Multimodal User Interfaces*, vol. 9, no. 1, pp. 17–29, Mar. 2015, doi: 10.1007/s12193-014-0161-4.

[24] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *Springerplus*, vol. 5, no. 1, Aug. 2016, doi: 10.1186/s40064-016-2941-7.

[25] "k-nearest neighbors algorithm," *Wikipedia*. May 11, 2020, Accessed: May 12, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=956126643.

[26] "Backpropagation," *Wikipedia*. Apr. 27, 2020, Accessed: Apr. 28, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Backpropagation&oldid=953508321.
[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.
[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[29] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," *arXiv:1808.01974 [cs, stat]*, Aug. 2018, Accessed: Apr. 11, 2020. [Online]. Available: http://arxiv.org/abs/1808.01974.
[30] D. E. King, "Max-Margin Object Detection," *arXiv:1502.00046 [cs]*, Jan. 2015, Accessed: Mar. 21, 2020. [Online]. Available: http://arxiv.org/abs/1502.00046.
[31] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, vol. 1, pp. 886–893, doi: 10.1109/CVPR.2005.177.
[32] A. T. Beck, R. A. Steer, R. Ball, and W. F. Ranieri, "Comparison of Beck Depression Inventories-IA and-II in Psychiatric Outpatients," *Journal of Personality Assessment*, vol. 67, no. 3, pp. 588–597, Dec. 1996, doi: 10.1207/s15327752jpa6703_13.
[33] M. Valstar *et al.*, "AVEC 2016 - Depression, Mood, and Emotion Recognition Workshop and Challenge," *arXiv:1605.01600 [cs]*, Nov. 2016, Accessed: Apr. 06, 2020. [Online]. Available: http://arxiv.org/abs/1605.01600.
[34] A. Ray, S. Kumar, R. Reddy, P. Mukherjee, and R. Garg, "Multi-level Attention network using text, audio and video for Depression Prediction," *arXiv:1909.01417 [cs, eess]*, Sep. 2019, Accessed: Apr. 06, 2020. [Online]. Available: http://arxiv.org/abs/1909.01417.