CS 890ES Project Report

on

Identification of 3D Attention Points in Virtual Reality Scenes

Sandeep Reddy,
MSc. Computer Science,
University of Regina
srt156@uregina.ca

Vineel Gannu,
MSc. Computer Science,
University of Regina
vgc361@uregina.ca

04/07/2020

Contents

1 Introduction

Attention points determine the interest of people based on where they look. It can provide important information in communication, such as giving cues of people's interest and attention, facilitating turn-taking during conversations, giving reference cues by looking at an object or person and indicating interpersonal cues such as friendliness or defensiveness [1]. It is crucial in many applications. It can help disabilities to use computers. The main use cases of attention points can be broadly categorized into (i)desktop computers (ii) television panels (iii) head-mounted (iv) automotive devices (v) handheld devices [2]. Desktop platform-based applications include the use of the eye gaze for computer control, text entry and communication and entering gaze-based passwords [3]. Attention points have also been used in television or video user interfaces. Head-mounted gaze tracking systems typically consist of two or three cameras mounted on a user-carried support system [2]. The attention points can also be applied in human-computer interaction.

The current growth in Virtual Reality technology is happening at incredible speed. A common problem with Virtual reality environments is that while the visual imagery is usually very convincing, the spatial interaction methods are often unsatisfying [4]. These interaction methods can be replaced by eye gazing techniques that have intense applications. Finding the attention points in the virtual reality environment could also provide important information with applications including, but not limited to, advertisement, physiotherapy, and for efficient and memorable educational training.

2 Problem Statement

Anxiety disorders are a category of mental disorders that are marked by extreme anxiety and fear feelings [5]. It is characterized by feelings of intense and persistent anxiety, fear, and nervousness. According to the 2014 Survey on Living with Chronic Diseases in Canada (SLCDC), approximately 3 million Canadians (11.6 percent) aged 18 or older reported having a mood and/or anxiety disorder in 2013. As stated by Anxiety and Depression Association of America (ADAA), Anxiety disorders are the most prevalent mental condition in the United States, affecting 40 million people aged 18 and older in the United States, constituting 18.1 percent of the population per year.

In general, anxiety treatment involves therapy and psychological counseling. Cognitive behavioral therapy (CBT) is a type of psychological treatment that is effective for a variety of issues including depression, alcohol usage, and anxiety disorders. CBT helps people to reduce stress, anxiety and face day-to-day challenges in life. Virtual reality based cognitive behavioral therapy (CBT) has wider advantages that can help to alleviate fear and anxiety.

3 Solution Overview

The solution includes the creation of a virtual environment (VE) with two different images on a wall that depicts two different emotions. An eye tracker is used to track the eye movement of the user/participant. VE is displayed for various participants and they are asked to gaze at the wall. The attention points of the participants are then recorded along with various other features such as the position, orientation, scale of the images and other objects in the VE and then the dataset is created. A multi-layer perceptron (MLP) neural network is created and is trained with the collected dataset. The trained model can be used to predict the attention points of a participant in a VE with unknown features.

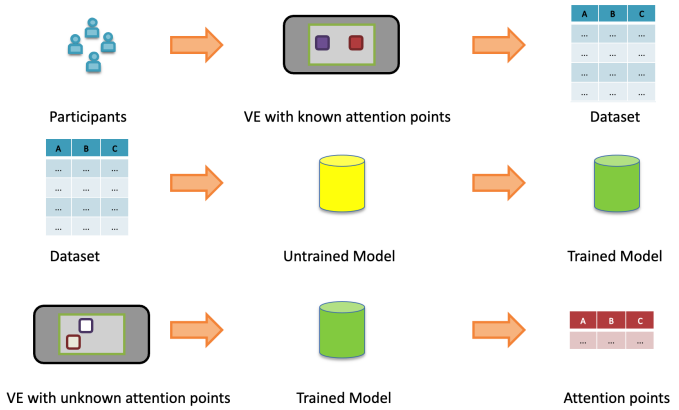

Figure 1: Solution Overview

4 Data

Six participants are involved to create the dataset. A virtual environment is created using unity software. The virtual environment has a plane and a wall with two images of two different emotions placed at different positions. The virtual environment is displayed to each participant and the participant is asked to gaze at the wall. Tobii eye tracker is used to capture the gaze positions of the participant. 1000 instances are created with various features such as the gaze position of the participant, position, orientation, and scale of different objects in the virtual environment. We wrote a script in the unity that is used to capture the data. The script writes the data into a text file. Since the data is unstructured, we used Microsoft Excel to structure the data. The data is stored in the excel format and Google Drive is used to store the data.

object1position_x	object1position_y	object1position_z	object1rotation_x	object1rotation_y	object1rotation_z	object1scaling_x	object1scaling_y	object1scaling_z	object2position_x	object2position_y
6	2.7	-0.2	0	0	0	150	150	-5.5	2.5	
-9.2	1.8	-0.4	0	0	0	150	150	150	0.0	3.5
3.1	3.2	-0.2	0	0	0	150	150	150	7.4	2.7
-6.7	3.0	-0.3	0	0	0	150	150	150	-2.3	2.0
2.0	1.6	-0.2	0	0	0	150	150	150	-7.9	3.4
0.2	1.6	-0.5	0	0	0	150	150	150	4.7	1.9
-5.9	2.1	-0.2	0	0	0	150	150	150	2.2	3.2
-3.1	3.5	-0.5	0	0	0	150	150	150	-10.0	3.2
-4.9	3.1	-0.3	0	0	0	150	150	150	7.6	2.4
-0.3	1.5	-0.4	0	0	0	150	150	150	-2.3	2.6
1.8	2.1	-0.4	0	0	0	150	150	150	-4.9	3.5
1.0	2.6	-0.4	0	0	0	150	150	150	4.5	2.8
-4.2	2.5	-0.5	0	0	0	150	150	150	0.8	2.2
-7.7	2.7	-0.4	0	0	0	150	150	150	-2.0	2.0
0.1	3.3	-0.4	0	0	0	150	150	150	7.8	2.2
1.5	1.9	-0.3	0	0	0	150	150	150	-0.4	3.2
-4.7	3.3	-0.3	0	0	0	150	150	150	-6.6	3.3
-8.1	3.2	-0.5	0	0	0	150	150	150	6.2	2.2
-0.8	2.6	-0.3	0	0	0	150	150	150	-5.8	2.6
-4.8	1.8	-0.3	0	0	0	150	150	150	-7.8	2.3
4.0	1.8	-0.3	0	0	0	150	150	150	7.0	2.9
-9.8	3.2	-0.5	0	0	0	150	150	150	-9.0	1.7
-5.7	1.9	-0.3	0	0	0	150	150	150	-4.2	3.5
-9.6	2.2	-0.5	0	0	0	150	150	150	-9.4	3.5
1.1	3.2	-0.2	0	0	0	150	150	150	4.0	2.9
-6.7	2.9	-0.5	0	0	0	150	150	150	-10.3	2.1
-4.9	3.2	-0.2	0	0	0	150	150	150	1.1	1.6
6.7	2.3	-0.2	0	0	0	150	150	150	7.5	2.5
5.3	2.2	-0.2	0	0	0	150	150	150	2.3	2.8
-9.2	2.3	-0.4	0	0	0	150	150	150	5.2	3.1
1.6	2.9	-0.4	0	0	0	150	150	150	-3.1	2.0
-8.8	1.8	-0.5	0	0	0	150	150	150	-7.0	2.1
2.0	3.3	-0.4	0	0	0	150	150	150	-0.7	2.0
-3.9	1.9	-0.5	0	0	0	150	150	150	6.8	3.3
-3.4	2.6	-0.5	0	0	0	150	150	150	6.7	2.9
-7.8	1.9	-0.2	0	0	0	150	150	150	6.4	1.5
1.2	1.9	-0.2	0	0	0	150	150	150	-3.8	3.0
-1.9	2.9	-0.5	0	0	0	150	150	150	6.2	2.5
7.7	2.4	-0.4	0	0	0	150	150	150	7.4	1.9
6.4	1.6	-0.3	0	0	0	150	150	150	-6.6	2.2

Figure 2: Sample Data in Excel format

```
writer.WriteLine("Object - " + gameObjects[0].tag);
writer.WriteLine("Position - " + gameObjects[0].transform.position);
writer.WriteLine("Rotation - " + gameObjects[0].transform.rotation.eulerAngles);
writer.WriteLine("Scale - " + gameObjects[0].transform.localScale + "\n ");

writer.WriteLine("Object - " + gameObjects[1].tag);
writer.WriteLine("Position - " + gameObjects[1].transform.position);
writer.WriteLine("Rotation - " + gameObjects[1].transform.rotation.eulerAngles);
writer.WriteLine("Scale - " + gameObjects[1].transform.localScale + "\n ");

writer.WriteLine("Gaze Points:" + gameObjects[2].GetComponent<AttentionPointsPlotter>().gaze_

writer.WriteLine("Camera position-:" + gameObjects[4].transform.position);
```

Figure 3: Script in unity that writes the data to a text file

5 Tools

We used the following tools to accomplish the project.

- *Unity:* Unity software (version 2019.3.7f1) is used to build a virtual environment. We used unity personal edition which is completely free to use.

Figure 4: Unity

- *Tobii eye tracker:* Tobii Eye Tracker 4C is used to track the gaze point of the participant. The eye tracker is attached at the bottom of the screen and is connected to the computer/laptop via USB.

Figure 5: Tobii Eye tracker

- *Nvidia Titan V and 2080 Ti RTX Graphic cards:* Nvidia Graphic cards are used to offload from CPU for higher performance and increased productivity.

Figure 6: NVIDIA Graphic cards

- *Python:* We used python programming language to build the model and to explore and visualize the data.

Figure 7: Python

- *Jupyter Lab:* Jupyter Lab software is used to run and execute the python code. Documents and activities integrate with each other in Jupyter lab, enabling new collaborative computing workflows.

Figure 8: Jupyter Lab

6 Timeline

The following timeline is followed to accomplish the project.

Feb 2020 Creation of a Virtual Reality scene)

Feb 2020 Tobii eye tracker integration with Unity

Feb 2020 Capturing Dataset

Mar 2020 Creating a multilayer perceptron neural network

Mar 2020 Training the neural network

Mar 2020 Testing with the test data and evaluating the performance

7 Expected Outcomes

7.1 Minimum Viable Project

- As a minimum viable product, a virtual environment is built and 1000 instances of the dataset are created with different features such as the position, orientation, scaling of various objects in the virtual environment along with the gaze position of the participant.

7.2 Target Goal

- The goal of the project is to create a 3D attention points dataset and train a multilayer perceptron model which then predicts the attention points in a virtual environment with an unknown gaze position.

8 Data Analytic Lifecycle

Data Analytics Lifecycle is a methodology that involves six different phases: Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, Operationalize.

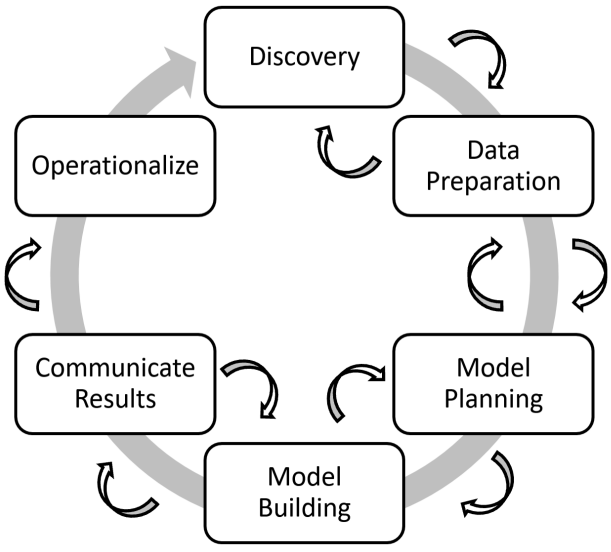

Figure 9: Data Analytic Lifecycle

8.1 Discovery

We started exploring various business issues that can be solved using data science. One of the issues that interested us is the treatment of Anxiety Disorder. There are few machine learning prediction models that are used to predict the anxiety disorder of a patient but treating the anxiety disorder using the machine learning model is novel. So we wanted to build a model that predicts the attention points of a user in a virtual environment which can be used to treat the anxiety disorder through Cognitive Behavioral Therapy (CBT).

8.2 Data Preparation

To create the dataset, we first created the virtual environment with two different images on a wall that depicts two different emotions. We used two sets of three different images that represent happiness and neutral emotions. Six participants are involved to capture the data. For each participant, the virtual environment is displayed. Based on the participant's input (on click of space button), the images are randomly spawned at different positions on the wall and the gaze position is captured. We created a script in unity that captures the data of various features of a virtual environment such as the position, orientation, and scaling of the objects in the virtual environment as well as the gaze position of the participant. 225 instances are captured and the data is unstructured. We structured the data in the excel format. Google Drive is used to store the data.

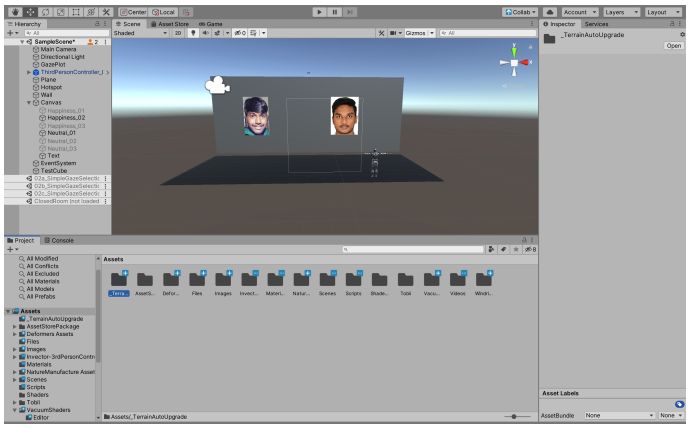

Figure 10: Unity in developer mode

Figure 11: VE displayed to participant

Figure 12: Sample code that captures data

8.3 Model Planning

Our project deals with the multi-target variables. For multi-target variables, we can either use multilayer perceptron or multi-target regression using clustering and decision trees. We chose a multilayer perceptron over multi-target regression because of its flexibility and ease of implementation with the help of current standard libraries such as keras and tensor flow. To train a multilayer perceptron, we need to preprocess the data that can be done by data normalization. Once the data preprocessing is done, the weights in the multilayer perceptron neural network can be initialized using Xavier/Glorot or He Initialization. The weight initialization helps in preventing the activation outputs from vanishing during the forward propagation in the neural network. We need to choose an activation function which that determines the output of a neuron based on the input. Relu, sigmoid, tanh are few common activation functions that are highly used. Batch normalization is a technique that is followed in the neural networks that standardize the input for every batch. It stabilizes the neural network and also improves speed and performance. When you train the model and if the model overfits the data, we can use dropout, which is a regularization technique that reduces the overfitting in neural networks. Optimizers are used in the neural network, to measure how wrong the model predicts and the parameters can be modified accordingly. Hyperparameters are the parameters whose values are set before model training begins. In neural network architecture, these parameters are the number of neurons and the number of layers that are to be set before the training. For the dropout, the hyperparameter is the dropout rate that is to be specified so that the model drops out the neuron based on the dropout rate. In optimizers such as Adam, beta1, beat2 which are the exponential decays and the learning rate acts as the hyperparameters. Loss functions are the functions that optimize the parameter values in the neural network and is also used to compile the model. Based on the type of business problem, the loss function is chosen. If it is a 2-class classification problem, then logarithmic loss function is chosen, whereas if the problem is k-class and regression, multi-class logarithmic loss, and squared error are used respectively. When large updates are made to the weights during training, the chances of overflow or underflow are very high which is known as exploding gradients. Exploding gradients can be prevented using gradient clipping. As discussed, dropout is one of the techniques to avoid overfitting. There are other methods such as cross-validation, removing features, training the model with more data, which can be used to prevent overfitting. Keeping these points in mind, we planned to build and train the model. If required, few of the above-discussed points can be excluded to build a high-performance model.

8.4 Model Building

After selecting Multilayer perceptron as our model we need to deal with a few hyperparameters and choose some functions and algorithms that fit to build the model for the data. We have decided to have one input layer, two hidden layers, and one output layer after performing some sets of training. As we know MLP updates weights using backpropagation technique i.e chain rule + memorization. We need to initialize the initial weights with the Xavier rule because we are going to use the tanh activation function. Updating weights operation is performed till the convergence so, for faster convergence and for high performance we chose adam optimizer. As we try to compile the model it seems to be overfitting. Then we added the dropout layer between each layer of the model. Dropout is used for regularization which is used to overcome the model from overfitting. For the adam optimizer, dropout rate and $\beta_1, \beta_2 are initialized with the standard values that are used in most of the real - world applications.$

```
# MinMaxScalers for features and output labels
scaler_x = MinMaxScaler()
scaler_y = MinMaxScaler()

# Fit and Transform the data using MinMaxScaler

file[features] = pd.DataFrame(scaler_x.fit_transform(file[features]))
file[outputlabels] = pd.DataFrame(scaler_y.fit_transform(file[outputlabels]))

X_test= pd.DataFrame(scaler_x.fit_transform(testfile[features]))
y_test= pd.DataFrame(scaler_y.fit_transform(testfile[outputlabels]))
```

Figure 13: MinMaxScaler for Data Normalization

```
model = Sequential()
model.add(Dense(30, input_dim=30, activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(16,activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(8, activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(3, activation='tanh'))
```

Figure 14: Neural network layers along with the dropout

We created 1000 instances of the dataset to input for the model. 560 instances are used to train the model whereas 240 are used to validate and 200 are used to test the model. We trained the model with 350 epochs. We used MinMaxScaler to normalize the input data. We used various activation functions such as Relu, tanh, sigmoid and tanh activation function gave better results when compared to other two.
The hyperparameters that we used in our project are:

- Number of layers - 4
- Number of neurons - (30,16,8,3)
- Dropout rate - 0.5
- Learning rate=0.0001
- beta 1=0.88
- beta 2=0.911

We used Adam optimizer that resulted in better performance. Since our project deals with regression, we used mean squared error as the loss function.

```
# compile the keras model
opt=Adam(learning_rate=0.0001, beta_1=0.88, beta_2=0.911, amsgrad=False)
model.compile(loss='mean_squared_error', optimizer=opt, metrics=['mse'])
```

Figure 15: Adam optimizer

The model that we built has an input layer, two hidden layers, and an output layer. The input layer receives 30-dimensional input. Altogether, there are 1,589 params. We used dropout to reduce the over-fitting of the data. We visualized the data by plotting the distribution of normalized data of positions of object 1 and 2 which are the position of images on the wall and also we visualized the correlation matrix that determines the correlation among features.

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 30)	930
dropout_4 (Dropout)	(None, 30)	0
dense_6 (Dense)	(None, 16)	496
dropout_5 (Dropout)	(None, 16)	0
dense_7 (Dense)	(None, 8)	136
dropout_6 (Dropout)	(None, 8)	0
dense_8 (Dense)	(None, 3)	27

Total params: 1,589
Trainable params: 1,589
Non-trainable params: 0

Figure 16: Model Summary

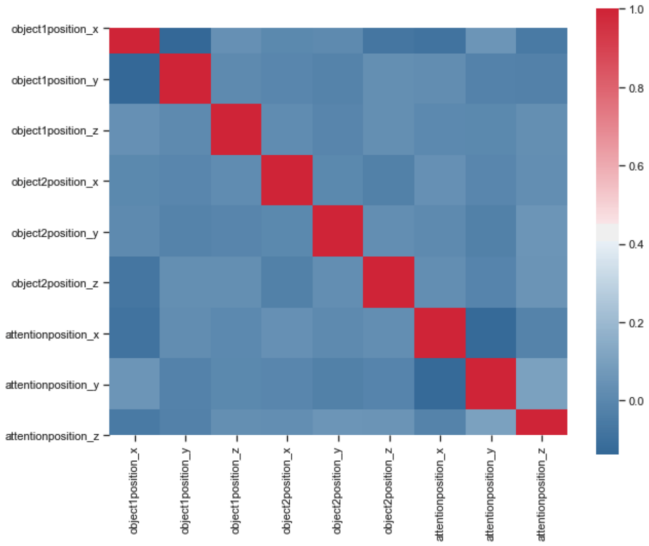

Figure 17: Correlation Matrix

8.5 Communicate Results

The trained model is tested on 200 instances. There is a convergence in the train and test loss as epoch increases. When the model is trained on 350 epochs, the training loss is 0.0262 whereas the test loss is 0.0202. As the number of epochs increases, the training and test loss decreases. When the training and test loss converge, we stop training the model, which prevents overfitting. If our project was implemented in an organization, we would have communicated results with the members within the organization. The model was overfitting when we trained it with 200 instances. After training it with 560 instances and using the dropout technique has reduced the overfitting.

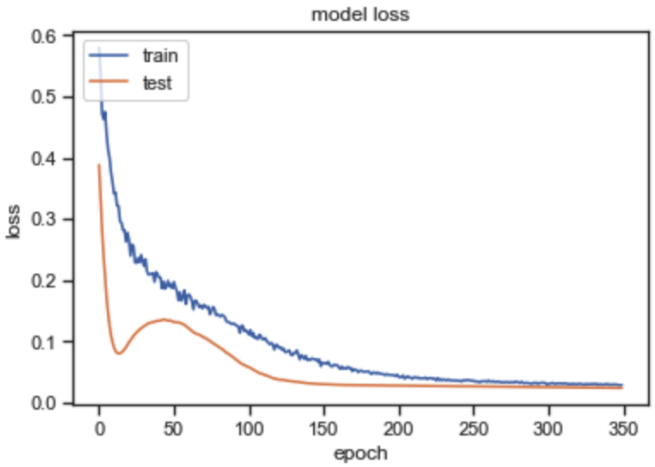

Figure 18: Model Loss

8.6 Operationalize

In this phase, we can deploy the project to the production environment. Instead of deploying the project on a wide-scale, we can deploy it on a smaller scale which allows the team to manage risk effectively. When the project is deployed to the production environment, the team monitors the performance and accuracy of the model and make necessary changes.

As a part of this project, we created a virtual reality environment that has two different images depicting two different emotions. With the help of six participants, we captured 1000 instances of the dataset that has object position, object scale, gaze position as different features. We used this dataset to train a multilayer perceptron neural network. Firstly, in the data preprocessing stage, we normalized the data using MinMaxScaler. We used tanh as the activation function that determines the output of the neuron based on its input. Initially, when we trained the model, the model overfits the data. To reduce the overfitting of data, we used the dropout regularization technique. Since the business problem of our project falls under regression, we used mean squared error as the loss function. Out of 1000 instances that we captured, we used 560 instances to train the model, 240 instances to validate the model and 200 instances to test the model. The model created can now be used to predict the attention points of a participant in a virtual reality scene that has a wall with two different images of different emotions.

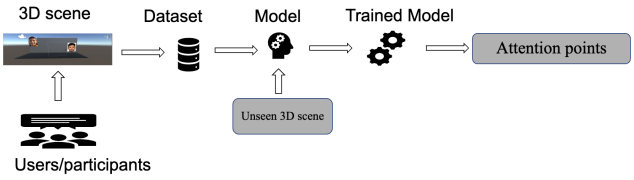

Figure 19: Process of Identifying the Attention points

Attention points can be used in Attention Bias Modification, a technique that is used to treat the anxiety disorder. It is designed in such a way that the attention of the anxiety disordered patients is shifted away from the threat. It can also be used in virtual reality therapy. Virtual Reality therapy has proved particularly successful when it comes to treating post-traumatic stress disorder (PTSD). Virtual reality therapy along with the attention points is used to diagnose and treat the psychological disorder of patients. In VR-based therapies, the virtual world provides an artificial, controlled stimuli along with the therapist able to monitor the reaction of the patient. For each patient's reaction, the therapists can determine the triggers and triggering levels. Replaying the virtual reality environment along with the eye tracker allows patients to habituate to such environments which in turn reduces anxiety.

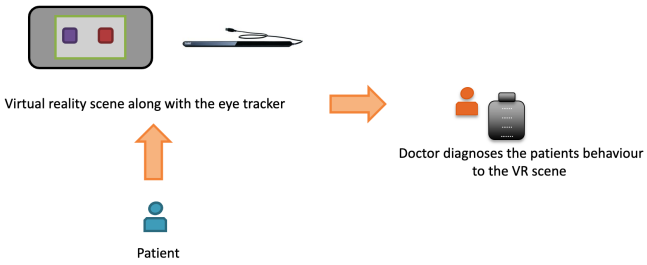

Figure 20: Virtual reality therapy

9 Group Members and Roles

9.1 Sandeep Reddy

- Sandeep Reddy worked on Data Cleaning, creation and training of multilayer perceptron model and documenting the project slides. Sandeep has an year of IT experience as a systems engineer with no prior experience in data science.

9.2 Vineel Gannu

- Vineel Gannu is responsible for creation of virtual reality environment along with the dataset, model evaluation and documenting the project report. Vineel has more than two years of IT experience as a pega senior developer with no prior experience in data science.

10 References

[1] Kang Ryoung Park, Juno Chang, Min Cheol Whang, Joa Sang Lim, Dae-Woong Rhee, Hung Kook Park, and Yongjoo Cho. Practical Gaze Point Computing Method by 3D Position Estimation of Facial and Eye Features

[2] Anuradha Kar. A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms

[3] Hyrskykari, A., Majaranta, P. and Räihä, K.-J. From Gaze Control to Attentive Interfaces. Proc. HCII 2005, Las Vegas, NV, July 2005

[4] Robert C. Zeleznik Andrew S. Forsberg Jurgen P. Schulze Look-That-There: Exploiting Gaze in Virtual Reality Interactions.

[5] Statistics Canada, 2013 Canadian Community Health Survey: Annual Component. (https://www.canada.ca/en/public-health/services/publications/diseases-conditions/mood-anxiety-disorders-canada.html)

[6] Yuko Hakamata, Shmuel Lissek, Yair Bar-Haim, Jennifer C. Britton, Nathan A. Fox, Ellen Leibenluft, Monique Ernst, and Daniel S. Pine. Attention Bias Modification Treatment: A Meta-Analysis Toward the Establishment of Novel Treatment for Anxiety

[7] Thomas E. Hutchinson, K. Preston White, JR., Senior Member, IEEE,Worthy N. Martin, Kelly C. Reichert, and Lisa A. Frey. Human- Computer Interaction Using Eye-Gaze Input

[8] Emmanuel G. Pintelas, Ioannis E. Livieris, Theodore Kotsilieris, Panagiotis Pintelas. A review of machine learning prediction methods for anxiety disorders

[9] Rizwan Ali Naqvi, Muhammad Arsalan, Ganbayar Batchuluun, Hyo Sik Yoon and Kang Ryoung Park. Deep Learning-Based Gaze Detection System for Automobile Drivers Using a NIR Camera Sensor

[10] Fred Stentiford. Attention-based vanishing point detection

[11] Stefan C. Schmukle. Unreliability of the Dot Probe Task