Empathy Training using Virtual Environments

Ron Jackson
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80907
rjackso7@uccs.edu

Sudhanshu Kumar Semwal
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80907
ssemwal@uccs.edu

ABSTRACT

We developed a virtual environment (VE) for nursing students so that they can experience what a person living with schizophrenia constantly hears. In our implementation, Non Player Character (NPC) Eva interacts with the player by recognizing the facial expressions of the players wearing Oculus Rift-S head mounted display (HMD). We use the Unity game development platform, and implement machine learning (ML) algorithms using deep learning (DL) models to provide such simulated experiences. In our implementation, the NPC Eva recognizes the player's facial expressions and reacts with a variety of facial, body, and verbal animated responses. Our empathy training virtual environment is developed for the nursing students. Our colleagues at the College of Nursing have also undertaken an approved IRB (Internal Review Board) user study. This paper focuses on technical details of our algorithms, their implementations. Main results of our research are summarized, including a positive reception of our empathy training virtual environment.

Keywords

Computer vision, facial expression recognition, neural networks, virtual reality, visual interaction, Python, Unity3D(TM)

1 INTRODUCTION

Our goal is to create a first person Virtual Environment (VE) providing a first person perspective, enabling a participant to experience auditory hallucinations [Deegan2022] a person living with schizophrenia might hear. Our VE provides an ability to move through, explore, and interact with places, objects and non-player characters (NPCs) to accomplish tasks that are consistent with the designed theme of the Nursing Curriculum. Our VE asks players (i.e. Nursings students in our case) to locate objects and interact with NPCs while visiting common VE locations such as a street-side coffee kiosk. NPCs interact with players using scripted behavioral responses.

Empathy is an essential skill in the nurse client relationship which can be trained and practiced [Ward2016, Ward2012, Ward2009] Even though empathy has been recognized as an important skill set, [Ward2016, Ward2012] report a decline in empathy in undergraduate nursing students. This may be attributed to time constraints in developing quality relationships with patients emphasizing the critical nature of empathetic interactions [Ward2009]. Our motivation to develop a VE is to provide an immersive VR training environment which can be an effective means of developing nursing students' empathy towards persons with schizophrenia as nursing students can also also experience the same auditory hallucination using our VE at any time of their convenience. Our implementation also addresses the partial occlusion of the user's face by the Oculus Rift S headmounted display (HMD) by providing a facial expression recognition (FER) DL model trained initially on full facial image, then refined by using only lower half facial images during final training by adding lower half of images avoiding upper half of these images which are covered by a VR headset in our case. We use Python-based FER application server to continuously communicate with the Unity VR platform client. The server uses an efficient residual network (ResNet) convolutional neural network (CNN) design which is trained to recognize five facial expressions. We used the Facial Expression Recognition Challenge 2013 (FER2013) data set [Carrier2013], Real-world Affective Faces (RAF) database, and some supplemental user images to train and test the CNN model [Jackson2022]. The resulting trained CNN provides 81% recognition accuracy across five different facial expressions on images with an occluded upper half face, providing

us a basis to show that our basic hypothesis works. In future, this recognition rate could be improved by adding more training data set. Figure 1 shows our basic setup. Additionally, the server-to-client inter-application communication average response time measured from client request to client receipt of server facial expression response is less than 0.25 seconds based on experimental observations (Figure 2). This responsiveness helps ensure timely and smooth initiation of NPC animation actions in our VE. The next sections of this paper discuss related work, design and implementation of our techniques. Eyes and eyebrows and folding features of our face are not visible due to VR-Headset (Figure 1) so the main technical challenge which we have addressed in this paper is how facial expressions can be computationally recognized by an NPC using only the bottom half of player's facial features as the player (nursing students) with headset would interacts with NPC. We implemented a VR scavenger hunt which replaces a real-life scavenger hunt experience designed in Nursing curriculum for all Nursing Students. Our system, which has undergone limited University approved IRB user study, is mainly geared towards recognizing player's facial expressions by an NPC to facilitate empathetic interactions. This may serve as part of online curriculum option for Nursing Students at our university, with the hope of developing empathy towards a person experiencing constantly denigrating and disturbing voices.

2 RELATED WORK AND MOTIVATION

We define empathy in the context of the nurse-patient relationship as predominantly a cognitive, rather than emotional, attribute that involves the ability to understand, rather than feel, experiences, concerns, and perspectives of the patient, combined with a capacity to communicate this understanding [Jackson2022]. Based on interactions with Nursing Faculty, it was thought that Nursing students would develop empathetic behaviors using our Virtual Environement where we can design storyline and scenarios where Nursing student player can experiences relatable experiences, such as hearing voices which a person with schizophrenia constantly experiences [Ward2012]. As clinical experiences are limited and may not present opportunities for real-life interactions with a person living with schizophrenia, our virtual environment can provide an online option for training Nursing students to experience relatable experiences. Schizophrenia is a serious mental wellness issue that interferes with a person's ability to think clearly, manage emotions, make decisions, and relate to others [Jackson2022]. It is a complex, long-term mental wellness issue, affecting about one percent of the population. Although schizophrenia can occur at any age, the average age of a person diagnosed with schizophrenia, and its onset, tends to be in the late teens to the early twenties for men, and the late twenties to early thirties for women. It is uncommon that schizophrenia is diagnosed younger than 12 or older than 40 years of age. The symptoms of schizophrenia fall into three categories: positive, negative, and cognitive [Jackson2022]. Positive symptoms are psychotic behaviors not generally seen in general population. People with positive symptoms may lose touch with some aspects of reality. Symptoms include hallucinations, delusions, thought disorders (unusual or dysfunctional ways of thinking), movement disorders (agitated body movements) [Jackson2022]. Negative symptoms are those that are associated with disruptions of normal emotions and behaviors, and include flattening affect such as reduced expression of emotions via facial expression or voice tone; reduced feelings of pleasure in everyday life, difficulty beginning and sustaining activities, and reduced speaking. Cognitive symptoms of schizophrenia can be subtle for some, but are more severe for others, and patients may notice changes in their memory or other aspects of thinking. Symptoms include possible poor decision making due to changes in the ability to understand information and use it to make decisions, trouble focusing or paying attention, and problems with working memory such as the ability to use the information immediately after learning it.

Virtual Reality based environments can be particularly valuable to develop critical skills and enhance the cognitive understanding when the same firsthand real world experiences are not available to experience, or are not safe [Jackson2022] as some nursing students may have apprehension towards working directly with a person with schizophrenia lacking training and skill set. Our Virtual environment may help a nursing student develop cognitive maps and experiences while training in a safe Virtual environment. Virtual environments can enhance such experiences by developing cognitive maps so that real-life experience of others can be experiences by augmenting a VE with hearing voices which a player can experience. As part of their course curriculum, the College of Nursing uses an audio recording of Hearing Voices that are Distressing [Deegan2022], and conduct a workshop to train students while they are asked to perform real world in-person tasks at different stations. The nursing students are our intended target as players in our Virtual Environment. Existing College of Nursing curriculum involvs listening to the hearing voices recording through audio earphones throughout the training session by the nursing students. During Scavenger hunt training, nursing students visiting different stations to finding objects, reading an article, taking a quiz, and solving crossword puzzles, all the time listening to the distressing hearing voices [Deegan2022].

Our immersive VR training environment provides similar training experiences of a scavenger hunt, for example, while immersing the user in the hearing voices audio recording. Cognitive maps [Tolman1948] are a series of psychological transformations that allow an individual to acquire, code, store, recall, and decode information about their spatial environments by developing an understanding of relative locations and attributes of the phenomena in their spatial environment [Tolman1948, Jackson2022]. [Deegan2022] hearing voices simulation is key component of our Virtual Environment and is used to provide an opportunity to develop a sense of empathy for those who suffer with schizophrenia. Recent VR research indicates the most important user experience mechanisms are illusion of virtual body ownership (user perception that the virtual body is their own), and agency (user attribution of actions in the VE as their own) [Barbot2020].

3 METHODOLOGY AND IMPLEMENTATION DETAILS

We added a visual interaction mode between the user and primary NPC (Eva) in our VE enabling the possibility of new user experiences. Figure 2 shows overall flow in implemented Virtual Environment. NPC-Eva's has the computational ability to see and recognize the approaching user's facial expression in one of five categories– angry, disgust, sad, happy, surprise– and then react with unique scripted animated facial, body and verbal responses. This allows us to simulate both storylines as planned – (i) user-initiated interactions, such as approaching or speaking, with Eva while getting in line at a Cofeee-kiosk with other NPCs, and (ii) NPC-Eva initiated interactions as NPC-Eva can see and respond to participant's facial expressions as the player approaches the coffee kiosk in our planned interactions (Figure 2). Eva's Faecial Expression Recognizer (FER) is a Python-based facial expression (FE) prediction server operating on player's (i.e. nursing student or one of the co-authors) facial image frames captured by a web camera. The server runs continuously and provides Facial Expression (FE) prediction replies to the Unity SE application client requests as the user initially approaches Eva at the coffee kiosk (See Figure 1). The visual interaction mode was the result of three main developments and provides the following functions:

- A custom frontal aspect detector trained to find and segment the user's HMD-occluded face is activated and defines a box bounding the user's face. The detector is developed using the DLib toolkit for image processing and ML tools [Dlib2021]. The custom detector model is a Support Vector Machine (SVM) using Histogram of Oriented Gradients (HOG) features trained on images collected of the author wearing the HMD.
- A ResNet CNN FER prediction model trained on five facial expressions (angry, disgust, sad, happy, surprise) given inputs that are HMD-occluded facial image frames predicts the user's expression (See Figure 2). The prediction model is a ResNet CNN developed using combined training images (16,329 total) from the Real-world Affective Faces Database (RAF) and the Facial Expression Recognition Challenge 2013 (FER2013) data sets [Carrier2013], plus a small set of supplemental training facial images (989 total) of the author. This prediction model performs recognition using only the lower-half of facial images and achieves 81% accuracy.
- A real-time message interface between the Python FE prediction model and the Unity SE application receives requests and provides predicted facial expression replies. The message interface uses ZeroMQ, which is an open-source messaging library supported in Python and by through NetMQ, which is a native port of the ZeroMQ [ZeroMQ2021] library. The messaging pattern used is Request/Reply, where the Unity SE application is the requesting client and the Python FE prediction script is the server providing string type facial expression replies whenever it receives a request. Timing data observed during experimentation shows that the total time between FER server replies when the SE application is making continuous requests typically ranges between 0.20 and 0.25 seconds.

The VE automatically sets Eva's response mode– positive or negative – based on NPC Eva's perception of the user's expression simulating relatable experiences [Jackson2022, Deegan2022]. Angry and disgust generate negative reactions, while sad, happy and surprise generate positive reactions. Figure 2 shows the overall functional flow of our Virtual Environment.

4 DEVELOPMENT AND COMPUTING PLATFORMS

The task pipelines for these capabilities, which are all implemented in Python version 3.7, are shown in Figure 3. The Unity and C development environments for this project are Unity version 2019.2.19f1 and Visual Studio Community 2019 version 16.9.3, respectively. The Oculus Application is version 28.0.0.222.469, using an Oculus Rift S HMD (Firmware version 2.2.0) and Touch Controllers (Firmware version 1.14.2).

4.1 HMD-occluded Facial Detection

The Python integrated development environment (IDE) for this project is Spyder 4.2.1 configured for Tensor-

Figure 1: Interaction set up and Eva's recognition of user facial expression and response

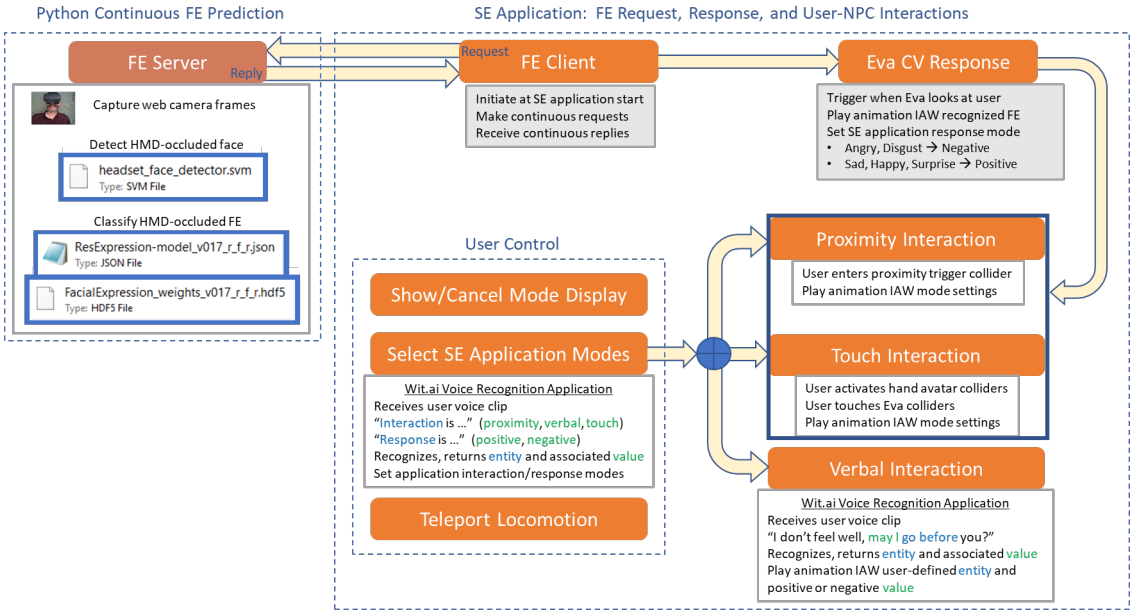

Figure 2: Functional flow in Implemented Virtual Environment (VE)

flow GPU support. The primary software library versions are: OpenCV - 2.4.1, DLib - 19.21.1, Tensorflow - 3.4.1, and PyZMQ - 20.0.0 (which supports the ZMQ library version 4.3.3). Hardware environment is a HP Omen Obelisk 875-0084 desktop computer with the following specifications: Intel® Core™ i7-9700 processor, 16 GB SDRAM memory, and NVIDIA® GeForce® RTX 2070 SUPER™ graphics card with 8 GB dedicated memory. The web camera used for image frame capture to implement computer vision is a Logitech ® C270 HD operating at 30 frames per second (FPS), with a frame width and height of 640 and 480 pixels, respectively.

The occluded facial detector developed in this project use Histogram of Oriented Gradients (HOG) feature descriptors and a support vector machine (SVM) model to perform detection classification. Both OpenCV and DLib provide SVM techniques for performing detection and classification using HOG and Haar Cascade feature descriptors. All feature descriptors seek to represent images, or grids of image patches, using a concise set of information that describes and distinguishes the content of the image. The resulting feature vector for the entire image will be much smaller in size than the total number of pixels and color channels of that image. A HOG descriptor is the distribution of two dimensional direction gradients of the image pixel intensities. The gradients are calculated for a dense, and overlapping, grid of pixel patches across the width and height of the image. Gradients values are larger where there are abrupt changes in pixel values, so these features are useful for detecting distinct parts of images such as corners and edges [Jackson2022]. The SVM [Mallick2018, Jackson2022] first attempts to transform the dataset under consideration so the classes within it can be separated linearly. Then, it solves a constrained optimization problem to find the best possible separating line distinguishing the classes. The best possible line is one that provides the largest margin between the line and the closest examples in either class, while minimizing the decision errors that are made [Jackson2022]. A data point that lies exactly on the margin boundary is a support vector [Mallick2018]. The description above

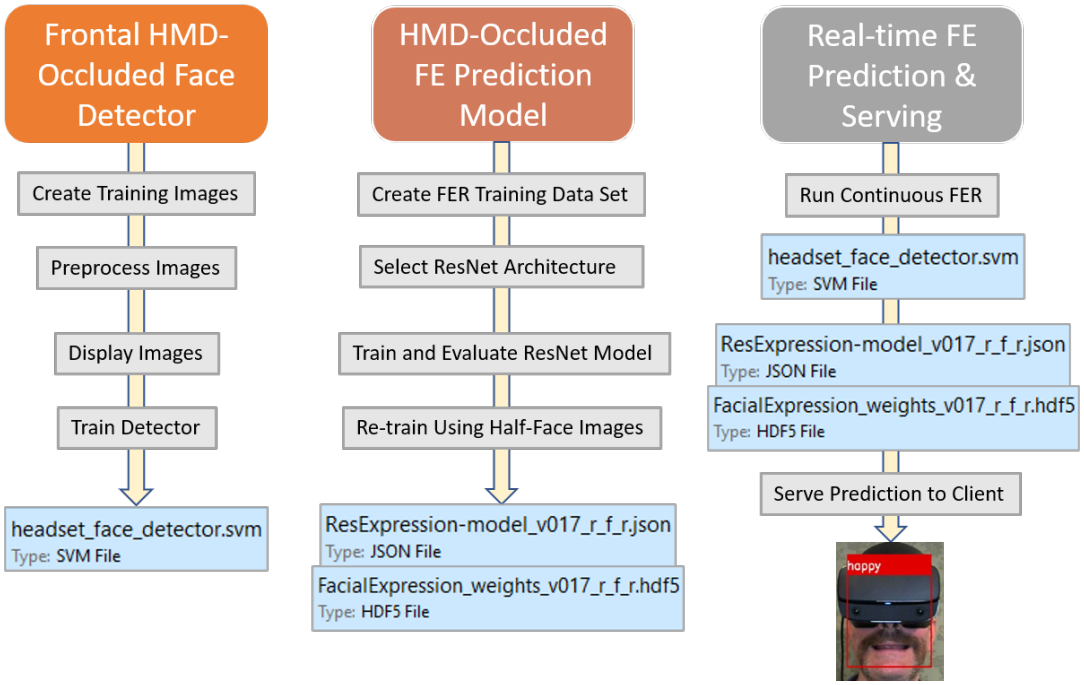

Figure 3: CV and FER Task Pipelines

is a 2D example. For higher dimensions, the SVM finds the best possible separating plane, or hyper-plane.

One of the challenges we faced was that images of the user's face occluded by the HMD were different that those not occluded in the data which we were using. So we needed to add faced with HMD to the data. These images were created using a web camera (Figure 4). A sliding image collection window captures user facial images at different locations across the web camera field of view. The window is outlined by a box when viewed in the web cam display. The user keeps their HMD-occluded face centered in the window as it moves. This is important because the coordinates defining the extent of the box sides will be used to specify the object to be detected during training of the HOG SVM. Figure 4 shows an example of training images displayed. Finally, the detection model is trained. The images are divided into 80/20% split training and test sets. The dlib.train_simple_object_detector() function trains and evaluates the model performance.

Figure 4: Training image samples displayed for verification

4.2 HMD-occluded FE Prediction Dataset

We decided at the outset of the project to implement a reduced set of facial expressions to demonstrate CV interaction in our implementation. Many of the data sets available for facial expression recognition have at least seven expression categories (for example, anger, disgust, sad, fear, happy, surprise, and neutral). Selecting only data corresponding to a subset of the full expression classes improves the overall recognition accuracy by eliminating a class that is hard to distinguish, for example, images labelled fear. Although not a neutral emotion, fear is often confused with several other expression classes in this implementation, and we felt that players (Nursing students) will welcome our story lines with excitement and curiosity and not with fear, as those were similar to regular training which nursing students undergo in their curriculum. Eliminating the neutral class also made sense because our story line handled both positive (smile) and negative (rude tone and statements) interactions as we did not plan any interaction for a neutral expression. As a result, we decided upon a facial expression set consisting of five – anger, disgust, sad, happy, and surprise – classification of player's lower half facial image. The remainder of this section explains the workflow tasks as shown in the middle column of Figure 3.

For the FER training, we chose existing data sets containing diverse images (many different subjects under varying conditions) as the core of the training set for this project. After researching the existing data sets, the RAF and the FER2013 data sets were selected. The RAF data set consists of color and monochrome images at a 100×100 pixel resolution. The FER2013 data set has monochrome images at a 48×48 pixel resolution. Some of the facial images have other objects (hands,

sunglasses, hats, watermarks, for example) that partially occlude facial features. Some images have varying aspect and rotation angles. We noticed that the existing data set composition across the five expression classes is unbalanced and is a common occurrence in these types of data sets. Finally, the FER2013 48×48 image pixel resolution is too small for the multiple ResNet convolution stages. These challenges are addressed by performing in-place data augmentation and class weighting during training of the CNN FER model. Both techniques help address the class imbalance inherent in the data sets. The combined data set uses RAF images from all five classes, with FER2013 images only from selected classes (disgust and surprise). This approach also helped to increase examples in the underrepresented classes, as that was our concern. Finally, the FER2013 image sizes were expanded to 100×100 pixels using interpolation and the number of channels was expanded to three for implementation.

Figure 5: Example RAF (bottom) and FER2013 (top) images [Carrier2013]

4.3 Occluded Facial Expression (FE) Prediction Model using ResNet Architecture

We used Keras with a Tensorflow backend to develop the ResNet CNN model. The code for the CNN models is implemented, trained and evaluated on Google Colab using available GPUs. The number of training epochs is explored experimentally (ranging from 30 to 120). GPUs were necessary to ensure reasonable model training times. After training and evaluating different model/training hyper-parameter configurations on Google Colab to identify the best performer, the selected model and weight files were moved to the Spyder development environment on the desktop computer for use in the real-time FE prediction server.

As explained in [Jackson2022], ResNet architecture provides an initial convolution layer (7×7, and 2×2 strides), up to three residual stage layers, average pooling and flatten layers, and a 5 node dense output layer. The convolution depth dimensions for the three ResNet stages are (64, 64, 256), (128, 128, 512) and (256, 256, 1024), respectively. Each residual stage contains three bottleneck convolution blocks where the first block uses a projection shortcut to match input and output dimensions. The remaining blocks use identity shortcuts. The total depth of this model (counting the initial 7×7 convolution and the final dense layer) is 29 (ResNet-29) if all three residual stages are used, and 20 (ResNet-20) if only the first two residual stages are used.

4.3.1 Training and Evaluating ResNet Model as facial expression classifier

The best performing model was designated ResExpression-model_v014_r_f.json (which also has an associated model weights file) with an average prediction test accuracy of 88%. Experimentation showed that training this model for more than 90 epochs did not significantly improve validation accuracy, but showed signs indicating the onset of over fitting (training loss continues to decrease, but validation loss stays the same, or increases).

Expre-ssion	Prec-ision	Recall	F1	Sup-port.
angry	0.70	0.78	0.74	131
disgust	0.65	0.78	0.71	221
sad	0.76	0.86	0.81	383
happy	0.96	0.89	0.92	925
surprise	0.97	0.92	0.94	798
Accuracy			0.88	2450
Macro Avg.	0.81	0.84	0.82	2450
Weighted Avg.	0.89	0.88	0.88	2450

Table 1: Full face model classification report

4.3.2 Retrain Selected Model Using Half Face Images

The final step to develop the HMD-occluded FE prediction model is adaptation of the full face model to a half face expression prediction model. This is done by retraining the full face model on the lower half of the images in the Reduced RAF+FER2013 data set. All images in the Reduced RAF+FER2013 were processed to remove the top 50 rows of pixels and then the images were resized using interpolation back to 100×100 pixels. The model was retrained using the same architecture, characteristics and training hyper-parameters as described in the previous section. This allowed us to train the model using the examples we collected from those wearing the HMD. The resulting model provides 80% prediction accuracy on the half-face test images, losing only 8% performance relative to the full face model. However, during experimentation using continuous image capture and FE prediction, we felt this model did not provide stable FE prediction over the amount of time necessary to provide a consistent input to the our empathy application module, and will need to be improved in future. The implemented solution to increase Facial Epression (FE) prediction stability was to introduce a relatively small (989 images, < 6% of the dataset total) set of user face supplemental images into

the Reduced RAF+FER2013 dataset. User facial images presenting all five expressions were captured and classified into the five different expression classes. An example of the user face supplemental images is shown in Figure 6.

Figure 6: User face supplemental image examples

The FE prediction model was retrained on the half-face version of this new data set (Reduced RAF+FER2013+RKJ, using the second author's initials to represent the supplemental images). A sample of the half-face images from this dataset is shown in Figure 7. The resulting model provides 81% prediction accuracy. This is only a 1% increase in prediction accuracy compared to the data set without the supplemental images, however, experimental observation indicates that the stability and consistency of FE prediction is significantly improved in real-time continuous image capture with the web camera. This half-face prediction model is designated ResExpressionModel_v017_r_f_r.json (and also has an associated model weights file, FacialExpression_weights_v017_r_f_r.hdf5). This is the model used to move forward in the project and implement the real time FE prediction server. Table 2 shows the overall classification report for half face images.

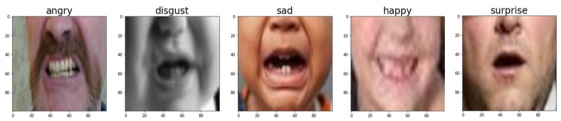

Figure 7: Reduced RAF+FER2013+RKJ half-face images

4.4 Real Time FE Prediction and Serving

Implementation of the Computer Vision based interaction requires the continuous capture of facial images while the user is wearing a VR-headset or HMD. The web camera is positioned directly in front of the user and at the height of the upper chest, or lower neck, as shown in Figure 1. The camera positioning ensures a full and clear view of the lower half face and provides the best performance as determined by experimentation.

Expre- -ssion	Prec- -ision	Recall Recall	F1 F1	Sup. Sup.
angry	0.56	0.72	0.63	157
disgust	0.56	0.68	0.61	271
sad	0.66	0.68	0.67	368
happy	0.90	0.89	0.89	945
surprise	0.93	0.82	0.88	857
Accuracy			0.81	2598
Macro Avg.	0.72	0.76	0.74	2598
Weighted Avg.	0.82	0.81	0.80	2598

Table 2: Half face model classification report

4.4.1 *Run Continuous FER*

Initialization tasks are first accomplished to prepare for continuous frame capture and FE processing. These tasks include: using OpenCV to assign the web camera as a video frame capture source and create a window for display of the frames, loading the DLib headset_face_detector.svm model, loading the Keras ResExpression-model_v017_r_f_r.json model and FacialExpression_weights_v017_r_f_r.hdf5 weights files, and setting up the ZeroMQ message socket/address to bind (connect) as the reply server. After performing the above initialization, the FER server enters into a continuous loop which can be terminated by the user through a keyboard command. Each pass through the loop: (a) Captures a web camera frame, (b) Detects HMD occluded faces using headset_face_detector.svm, (c) Extracts the portion of the frame corresponding to the HMD occluded face, (d) Creates a lower half image of the detected HMD-occluded face, (e) Resizes the image to 100×100 pixels, (f) Normalizes the image pixel values between 0 and 1, (g) Reshapes the image as a tensor, (h) Sends the image to ResExpression-model_v017_r_f_r.json for FE prediction.

The results of the prediction are displayed as an annotated box around the user's HMD occluded face in the display window upon completion of each pass through the loop. See Figure 8 for an annotated example of FE as surprise.

4.4.2 *Serve Face Expression (FE) Prediction to the Client*

FE prediction serving also occurs within the continuous loop described above. During each pass through the loop, the following actions are performed: (a) The reply server socket tries to check for the receipt of a request from the Unity request client, (b) If there is no request, an exception is thrown which is handled by printing a

message to console, (c) Execution flow is returned to the outer continuous FE prediction loop. (d) Else there is a request, so print the request to console and do the next four steps, (e) Append the most recent FE prediction to a current expressions list of maximum length of 10, based on the user selected value, (f) Determine the most common prediction in this list, (g) Reply to the Unity request client with the most common prediction, (h) Print the current expression list and reply to console, (i)Execution flow is returned to the outer continuous FE prediction loop.

Experimental observation of the time required to complete a request-reply cycle shows that the majority of these transactions occur within 0.20 - 0.25 seconds. Most of this time is used by the FE detection and prediction workflow. The additional time added by the message interface for requests and replies is within the range of 0.001 - 0.01 seconds.

Figure 8: HMD occluded FE prediction display window

4.5 SE Application FE Request Client

The SE application also required modification of the primary NPC controller, the addition of a threaded request client class, and new visual interaction animations for the primary NPC, Eva. The primary NPC controller is modified to implement both the continuous FE requests and to provide the animation triggers upon receipt of the FE predictions. The controller instantiates and activates the requester class and provides animation triggers upon receipt of FE predictions. The NPC controller initiates the visual interaction animation when the user is within range (a specified distance and angle) that also corresponds to the conditions necessary to trigger Unity Inverse Kinematics animation that causes Eva to "look" at the user. Based on the expression message received from the FE prediction server, one of five Eva animation responses are triggered. Additionally, the application response mode is set to positive if the the received expression is sad, happy, or surprised. The response mode is set to negative if the received expression is angry or disgust.

4.5.1 Facial Animation and Audio Clips used for NPC-Eva

We obtained body pose animation clips from Mixamo.com, and used these as NPC body animation building blocks for Eva. However, these clips do not provide facial animation. We added these expressions by first composing individual verbal responses for Eva to deliver when observing each user facial expression. The length of each spoken response was designed to match the duration of the corresponding body pose animation sequence. Then, facial expression animations were added to each body pose animation using the blend shapes available in the Eva Skinned Mesh Renderer. The Eva model provides many facial blend shapes that can be used to create any type of facial expression. Multiple facial blend shape characteristics were selected to create the desired effect. Key frames were created in the Unity Mecanim Curves display to position each desired facial expression at the correct time during animation. The key frames were connected using curve interpolation available in the Mecanim Curves tool. During development of a facial animation, the sequence can be run on the Eva model in Unity editor mode to monitor the effect of each change. Finally, the verbal response for each visual interaction was recorded as audio clip. The audio clip is played automatically upon start of each animation sequence by embedding an event at time = 0 of the animation sequence.

4.5.2 Lip Sync Animation for NPC-Eva

A final animation was incorporated which uses Eva's Viseme blend shapes to synchronize her mouth and lip movements to the playback of each recorded verbal response. Visemes are the patterns of lip and mouth movements that correspond to the basic phoneme sound utterances during speech. Oculus provides a Lip Sync plug in to Unity that maps a model's available visemes to a standard set of visemes determined by Oculus. Oculus also provides a utility that computes and stores the visemes for each audio clip to make the playback during animation more responsive. The Lip Sync animations are triggered by a method incorporated into the primary NPC controller.

5 RESULTS AND INTERACTIONS

Primary interactive NPC, Eva, provides many different reactions to the user based upon a variety of simulated senses that trigger interactions. Additionally, Eva can also provide both initial first impression reactions of the user approaching the coffee kiosk, and additional reactions as the user attempts to break in line.

Eva's initial reaction is based upon her recognition of the user's facial expression in one of five classes: anger, disgust, sad, happy, or surprise. This visual interaction mode is activated whenever the user approaches Eva within a distance and angular range specified in the SE application. As the user continues to move forward and breaks line in front of Eva, a final reaction by Eva is presented based on the current setting of the application's interaction mode and Eva's perception of the user's mood based on her recognition of the user's facial expression. There is at least one animated response for each combination of three interaction modes, such as proximity, spoken, and touch and Eva's two facial expression recognition based response modes (positive and negative) as explained earlier [Jackson2022, Deegan2022].

The current SE application version provides the user a full set of control mechanisms. The user controls the modified application interaction mode setting with voice commands. The voice commands and the user's spoken interaction utterances are enabled via buttons on the Touch Controllers. Once selected, the touch interaction mode can be employed when the user depresses either of the Touch Controller palm triggers and "touches" Eva to push her away with the user hand avatar. The player can also display the application mode setting, and then clear this display, with Touch Controller buttons. The user can also moves through the VE with a teleportation mode of locomotion. Teleportation is accomplished with the thumb joystick on the right Touch Controller. Rotation of the user's facing direction is accomplished with the thumb joystick on the left Touch Controller.

6 SCENARIO, STORY LINE, AND EMPATHY INTERACTIONS

We implemented the following scenario and story line: (a) A room setting pops up. (b) When the player happen to reach to a fountain where the person starts to experience very disturbing voice like "Go away from here, you are not supposed to come here" etc. and he will try to navigate back to map and this time he meets with a dog on his away. (c) Dog interaction is provided as the user experiences general disturbing voice/noise like "Go out of my way", "you are not meant to be here." These voices are from [Deegan2022]. (d) Player tries to scavenger hunt to a coffee kiosk and meets EVA. Eva would offer the player her place in line, and provide empathetic interactions, or not, based on player's facial expressions. (e) The player buys a coffee and communicates with coffee seller, (f) Finally, from coffee kiosk he tries to head towards clinic.

6.1 Voice based Empathy Interactions

The modified application uses Wit.ai cloud-based voice recognition to enable voice commands and interaction

Figure 9: Eva response based on recognition of user's facial expression

with Eva. The voice recognition is based on entities and values. Move aside, go away, get lost, and get out will generate a negative response from Eva when the application is in the spoken mode. The voice recognition application is trained to recognize various sentences or phrases that contain the above mentioned entities and values. Below are only a sample of examples of sentence and phrase variations that may be recognized are provided below: (a) Get out of my way, I ... (want to, need to) ... go first. (b) "May I go first, please?", (c) (I'm tired, I don't feel well) "... please allow me to go first." These verbal statement are understood by NPC-Eva and generate negative and positive empathetic response from NPC-Eva.

7 CONCLUSION

Our implementation demonstrates an efficient messaging interface between the Python FE prediction server and the Unity C#.NET client. The ZeroMQ library provides a lightweight approach for creating a communication interface that was easier to implement. The C# port of ZeroMQ, is easy to implement in a separate thread so continuous client requests can be made without interrupting Unity graphic rendering tasks.

In summary, we implemented our primary goal of creating a 3D Unity based Virtual Environment (VE) so that a player can experience the hallucinatory sounds which person with schizophrenia constantly hears. Six new NPC body, facial expression, and voice animations were integrated into the primary NPC, Eva, to support the new visual interaction mode. We showed that NPCs, such as Eva, can provide provide empathetic verbal interactions based on positive and negative interactions by the player wearing a VR headset.

An IRB study has been undertaken by College of Nursing Faculty at our university over several semesters, and early results are promising, and bode well for our future efforts.

8 FUTURE WORK

In future, we imagine extending the virtual environment to an Augmented Reality (AR) space enabling a multi-

person immersion experience with integrated interactive presentations; potentially using actors to add depth to the simulation by playing the roles of concerned family and friends of the person with schizophrenia. In addition, we would like to include neutral face recognition by NPC to provide better transition, feedback, and experiences to the player by NPC. In addition, eyes provide expressive emotive features in human-faces. As see through VR-headset become more available, we anticipate future incarnations of our work to incorporate see through headset facial images for training and better accuracy.

9 ACKNOWLEDGMENTS

We are thankful for College of Nursing Faculty Drs. Deborah Pina-Thomas and Lynn Philips for their input, undertaking the University IRB case study, and providing a simulated recording [Deegan2022] of what a person with Schizophrenia hears.

10 REFERENCES

[Barbot2020] James C. Barbot, Baptiste; Kaufman. 2020. What makes immersive virtual reality the ultimate empathy machine? Discerning the underlying mechanisms of change. Computers in Human Behavior 111 (2020), 106431. https://doi.org/10.1016/j.chb.2020.106431

[Carrier2013] Pierre-Luc Carrier and Aaron Courville. 2013. Challenges in Representation Learning: Facial Expression Recognition Challenge. https://www.kaggle.com/c/challenges- in-representation- learning- facial- expression-recognition- challenge/data

[Cortes1995] C. Cortes and V. Vapnik. 1995. Support vector networks. Machine Learning 20 (1995), 273-297.

[Dalal2005] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 05),Vol.1.886-893

[McAllister2020] Jodi; McAllister Margaret; Lazenby Mark Dean, Sue; Halpern. 2020. Nursing education, virtual reality and empathy? Nursing Open 7, 6 (2020), 2056-2059.

[Deegan2022] Patricia Deegan, Hearing voices that are distressing: A training simulation experience. National Empowerment Center, NARPA.org, Accessed 2022.

[Claudia2020] Claudia; Riva Giuseppe; Villani Daniela Di Natale, Anna Flavia; Repetto. 2020. Immersive virtual reality in K-12 and higher education: A 10-year systematic review of empirical research. British Journal of Educational Technology 51, 6 (2020), 2006-2033.

[Ward2016] Julia Ward. 2016. The empathy enigma: Does it still exist? Comparison of empathy using students and standardized actors. Nurse Educator 41, 3 (2016), 134-138.

[Ward2012] Julia Ward, Julianne Cody, Mary Schaal, and Mohammadreza Hojat. 2012. The Empathy Enigma: An Empirical Study of Decline in Empathy Among Undergraduate Nursing Students. Journal of Professional Nursing 28, 1 (2012), 34-40.

[Ward2009] Julia Ward, Mary Schaal, Jacqueline Sullivan, Mary E. Bowen, James B. Erdmann, and Mohammadreza Hojat. 2009. Reliability and Validity of the Jefferson Scale of Empathy in Undergraduate Nursing Students. Journal of Nursing Measurement 17, 1 (2009), 73-88.

[Johnson2014] Keith Johnson and Sudhanshu K Semwal. 2014. Shapes: A Multi-Sensory Environment for the B/VI and hearing-impaired community. In 2nd International Workshop on Virtual and Augmented Assistive Technology (VAAT) at IEEE Virtual Reality 2014. 1-6.

[Mallick2018] Satya Mallick. 2018. Support Vector Machines (SVM). https://learnopencv.com/support-vector-machines-svm/

[Jackson2022] Ron Jackson, Using the Unity Game Development platform to build Virtual Reality Schizophrenia Empathy training applications, pp1-107, Advisor: SK Semwal, MS Thesis, University of Colorado Colorado Springs, 2022.

[ZeroMQ2021] Zeromq.org. 2021. ZeroMQ: An open-source universal messaging library. April 27, 2021 from https://zeromq.org/

[Tolman1948] Edward Tolman, In Pschological Review, Cognitive Maps in Ants and Man, 1948, vol 55, pages 189-208.

[Dlib2021] Dlib.net. 2021. DLib C++ Library. Retrieved April 10, 2021 from http://dlib.net/