# Adaptive Interaction System for a Virtual Bartender with Emotion Recognition

Michail Bakalianos
*Department of Information Technology*
*Uppsala University*
Uppsala, Sweden
michalianos99@gmail.com

Vencel David Koczka
*Department of Information Technology*
*Uppsala University*
Uppsala, Sweden
koczka990@gmail.com

***Abstract*—This document presents the development of a virtual bartender system using the Furhat Robotics platform. The system is divided into two main sub-systems: a User Perception sub-system, which processes real-time video to detect affective states like valence and arousal using facial feature extraction and machine learning models, and an Interaction sub-system, which employs a rule-based approach to generate contextually appropriate responses. The system integrates ResNet-18 for emotion recognition and Furhat SDK for interaction management.**

***Index Terms*—Human-Robot Interaction, Emotion Recognition, Affective Computing, FurhatSDK.**

## I. Introduction

The development of socially intelligent robots capable of perceiving and responding to human emotions has gained significant attention in human-robot interaction research. This project focuses on designing an interactive system utilizing Furhat Robotics' social robot platform within the context of a virtual bartender. The system integrates emotion recognition with adaptive conversational behavior to simulate real-time human-robot interaction. By detecting emotional states such as valence and arousal through live video input and machine learning techniques, the system aims to enhance user engagement and interaction quality.

The project comprises two main components: a User Perception sub-system, responsible for emotion detection using ResNet-18 [1] for feature extraction and classification, and an Interaction sub-system, which employs a rule-based framework to generate contextually appropriate behaviors. Despite challenges like occasional emotion detection inaccuracies and system instability, the system demonstrated its potential to bridge the gap between technical functionality and user-centric design, contributing to advancements in emotion-aware robotics.

## II. Methodology

During the development of the following subsystems, a series of design decisions were made and iteratively refined as the project progressed.

### A. System Design

The overall architecture and integration of the two subsystems are illustrated in Figure 2. Our implementation incorporates interaction with a large language model (LLM) chatbot, specifically utilizing the Gemini API.


Fig. 1. Jack our Furhat bartender.

The LLM chatbot interfaces with the Interaction subsystem, which employs a virtual Furhat robot simulation. User speech input is captured and converted into text for further processing. Simultaneously, the User Perception subsystem continuously analyzes the user's emotional state.

The processed text and the extracted emotional data are transmitted to the LLM chatbot, which generates a dynamic response and determines the agent's mood based on the provided inputs.

Leveraging the generated response and mood data, the virtual Furhat robot presents the output in a more engaging manner. The agent's emotional expression is visually conveyed through the Furhat simulation, while the response is delivered in a natural, human-like voice.

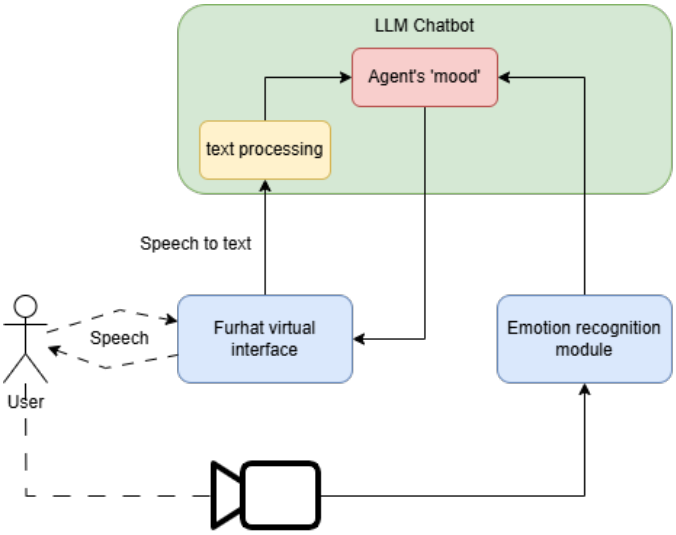


Fig. 2. Overall system architecture.

TABLE I
EMOTION COUNT IN THE DIFFUSIONFER DATASET

| Expression | Count |
|---|---|
| Neutral | 413 |
| Happy | 338 |
| Sad | 89 |
| Surprise | 166 |
| Fear | 73 |
| Disgust | 53 |
| Angry | 160 |

TABLE II
MODEL TRAINING RESULTS FOR EMOTION DETECTION

| Model | Learning Rate | Epochs | Test Accuracy |
|---|---|---|---|
| ResNet-18 | 0.001 | 15 | 73% |
| ResNet-18 | 0.0005 | 20 | 75% |
| MobileNet | 0.001 | 45 | 72% |

### *B. User Perception Sub-system*

The User Perception sub-system was developed to facilitate real-time emotion detection by analyzing facial expressions extracted from video input. The primary objective was to detect faces and accurately identify their emotional states. To accomplish this, we utilized the ResNet-18 convolutional neural network, known for its exceptional performance in feature extraction and classification tasks. The design prioritized modularity, ensuring smooth integration with the interaction sub-system while preserving the ability to process data in real-time. The implementation began with training ResNet-18 on the DiffusionFER dataset, a collection of synthetic images generated using generative artificial intelligence (GenAI).

For face detection, we employed the `py-feat` [2] library, which provided efficient methods to extract facial landmarks from video input. Initially, we attempted to use `py-feat` for detecting action units (AUs) related to facial expressions; however, this approach was found to be inefficient for our specific use case. Consequently, we transitioned to using ResNet-18 for more accurate and robust emotion classification. To process the face images, we resized them to 128x128 pixels, matching the training image size.

The model was trained on a GPU T4, allowing us to efficiently process the large dataset and train the model in a reasonable amount of time. During the training phase, we experimented with different configurations of ResNet-18 and also evaluated MobileNet to assess their performance. The table below summarizes the distribution of emotional expressions in the DiffusionFER dataset, highlighting its imbalance:

We evaluated different configurations of the ResNet-18 model and also experimented with MobileNet to see the impact on performance. The following table summarizes the results from various training setups:

The best model achieved a test accuracy of approximately 75% after training. Despite the promising results, there were several challenges. The small image size of 128x128 pixels, especially when using low-quality webcam images, made it difficult to achieve reliable predictions. Additionally, the synthetic nature of the DiffusionFER dataset presented a challenge, as the generated faces did not perfectly match the characteristics of actual human faces, which impacted the model's ability to generalize effectively. Furthermore, the quality of the webcam used in the real-time tests was sub-optimal, further complicating the emotion detection process. However, the results show that the system has potential for future improvements and optimization, and further refinements can be made to enhance its accuracy and robustness in real-world scenarios.

### *C. Interaction Sub-system*

The interaction sub-system employs a virtual Furhat robot, facilitating human-like communication between the user and our agent. This platform is capable of generating speech from input text and presenting predefined facial expressions.

To enhance the engagement of the interaction, we incorporated a large language model (LLM)-based chatbot (elaborated later) to generate the textual responses. Additionally, the mood of the Furhat robot was determined by the output of the LLM.

Prior to delivering a response, the Furhat robot conveys the intended mood through a sequence of brief, dynamic facial expressions (e.g., expressing annoyance by rolling its eyes). Each emotional state has 2-4 predefined expressions, from which the algorithm randomly selects one. After expressing the selected mood, the robot delivers the textual response using a predetermined voice.

The system implementation involved utilizing Furhat's Remote API to send control commands to the virtual robot. To select appropriate expressions, a Python function was developed that receives the robot's mood as input and randomly chooses a corresponding facial expression for that mood.

The interaction sub-system utilizes an LLM chatbot, implemented using the Gemini API. The chatbot was designed with a distinctive 'personality' inspired by a pirate bartender,

characterized by a slightly rude demeanor, quick wit, and a readiness to refuse service if users appear too intoxicated or disrespectful.

Using prompt engineering, we provided the LLM with both the user's mood and message. The LLM was instructed to return responses in JSON format containing three essential fields: the agent's response, the agent's mood, and whether the conversation had concluded. These fields enabled dynamic adjustment of the agent's mood, as previously described, and allowed for conversation termination when appropriate, facilitating the initiation of new dialogues.

## III. General Discussion

### A. Overall Pipeline

The complete flow of our system is illustrated in Figure 3. The two primary inputs are the user's speech and image. Speech is converted to text, while the user's mood is classified based on the image through the respective sub-systems. These extracted details are then sent to the LLM chatbot, which generates a response and determines the agent's mood. The agent's mood is processed by the expression selection function, which outputs a corresponding facial expression. The expression and response text are then transmitted to the interaction sub-system, which presents them in a human-like manner.

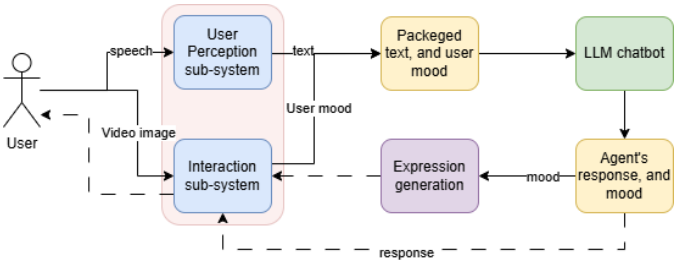


Fig. 3. The flow of information in the system.

### B. Challenges

The development of the User Perception sub-system faced several significant challenges, particularly related to hardware limitations and the training of the model. Due to the large size of the ResNet-18 model and the complexity of the dataset, training on our local machines was not feasible. As a result, we turned to Google Colab, utilizing the limited free quota and a T4 GPU for training. However, this setup was time-consuming, as the training sessions had to be frequently paused and resumed due to quota restrictions.

Another major issue stemmed from the synthetic nature of the DiffusionFER dataset, which led to discrepancies when applying the model to real-time webcam input. The synthetic images did not accurately reflect the variability of real-world scenarios, such as different backgrounds, glasses, or facial hair. These differences made it challenging for the model to generalize effectively, resulting in inaccurate emotion detection in real-world environments.

Finally, face detection proved to be a particularly resource-intensive process, often exceeding the memory capacity of our local machines and even occasionally crashing on the T4 GPU. The real-time demands of detecting faces, combined with the challenges of webcam quality, contributed to frequent failures in face detection, as well as misclassifications of emotional states. These issues underscored the limitations of current infrastructure and highlighted areas for improvement in future iterations of the system.

### C. Use of ChatGPT and Similar Tools

During the process of defining possible moods for our agent, we discovered that by specifying a limited set of moods, we could leverage ChatGPT to generate additional variations. This approach enabled us to expand the list of potential expressions for our Furhat agents more efficiently.

### D. Ethical Issues

One ethical concern in this project is the privacy of users' data, particularly since facial recognition and emotion detection are used. Although no personal data is stored, real-time processing of video inputs still raises privacy risks. Users' emotional states are analyzed without explicit consent for such deep analysis in casual settings. Additionally, the interaction sub-system includes a robot that engages in conversation, sometimes responding with a rude attitude. This behavior could be seen as inappropriate or offensive, raising concerns about the psychological impact on users, especially if the robot's demeanor is perceived as too harsh or inconsiderate in certain contexts.

Another issue is the potential bias in emotion detection, as the synthetic training data may not represent the full diversity of human facial expressions. This could lead to inaccurate or unfair classifications, particularly for underrepresented groups. Moreover, the robot's responses, influenced by the chatbot's personality, may perpetuate harmful stereotypes or contribute to negative user experiences if not carefully managed. Further refinement of both the emotion detection model and the chatbot's behavioral programming is necessary to ensure the system is ethical, inclusive, and respectful of all users.

## IV. Conclusion

This project demonstrates the potential of combining emotion recognition and interactive robotic behavior to create a socially intelligent virtual bartender. By integrating the User Perception sub-system, which leverages ResNet-18 for emotion detection, and the Interaction sub-system, powered by a large language model chatbot, the system provides an engaging and dynamic user experience. Despite challenges such as inaccuracies in emotion detection and limitations of synthetic training data, the system showcases promising results, with potential for refinement and future improvements in real-world applications.

Moving forward, improvements in hardware, model optimization, and data diversity are essential to enhance the accuracy and robustness of emotion detection. Additionally,

ethical considerations regarding privacy, user experience, and the potential for bias must be addressed to ensure the system is both inclusive and respectful. This work serves as a foundation for further exploration in the field of affective computing and human-robot interaction, with applications in a variety of socially intelligent systems.

Our implementation is publicly available for further exploration and use. The code can be accessed at the following GitHub repository.

## REFERENCES


[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv, 2015. [Online].

[2] J. H. Cheong, E. Jolly, T. Xie, S. Byrne, M. Kenney, and L. J. Chang, "Py-Feat: Python Facial Expression Analysis Toolbox," arXiv, 2023.