Real-Time Emotion Detection for Socially Assistive Robotics

Alondra Cárdenas | acardenas.2026@gmail.com
Windward School, Class of 2022
USC Viterbi Department of Computer Science, Interaction lab, SHINE 2021

Introduction

The field of socially assistive robotics aims to create robots that can engage with users through meaningful social interactions in order to provide non-physical, cognitive assistance [1]. In the Interaction Lab I worked with my labmate Luke and our mentor, Professor Matarić's Ph.D student, Brandon Thai Tran. With the help of our mentor, we wrote code to find faces via bounding boxes, perform facial landmark detection, train and run a machine learning model that detects human emotion based on the user's facial expressions, then displays the emotion on the video feed.

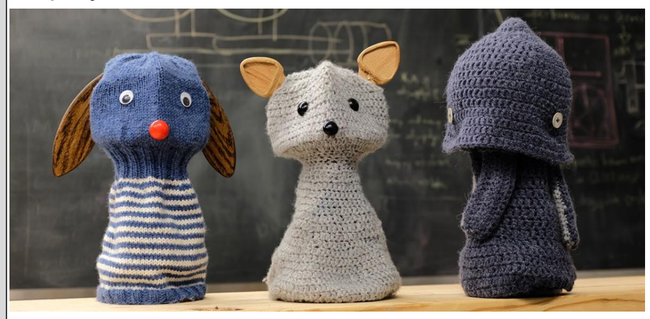

Figure 1. *3 examples of the socially assistive robot Blossom that we interacted with during the program and read research papers about* https://news.cornell.edu/stories/2019/05/soft-social-robot-brings-coziness-home-robotics

Objective & Impact of Professor's Research

Professor Matarić's lab is focused on designing and improving socially assistive robots (SARs) to help people, especially children, and those with special needs. SARs have the potential to help vast amounts of people including children with autism spectrum disorder (ASD), the elderly, those in convalescent care, and vulnerable populations in general [1]. SARs are meant to be an inclusive, effective, and accessible tool for the general population to provide support through human-robot interaction (HRI), machine learning, and robotics.

Skills Learned

I worked on writing a program and training a machine learning model to analyze visual data from live webcam data. The model would detect and identify a user's emotions in real-time based on facial landmarks. Through this research and the SHINE program, I practiced my programming skills through Python and MATLAB, learned to manage a coding environment through Poetry, developed clean coding practices, learned to effectively read a research paper, and used the terminal for the first time. I also gained experience using many Python packages like Pytorch, OpenCv, and Numpy. Additionally, I studied machine learning, SARs, and discovered what it means to be an HRI researcher. I also learned to utilize and navigate JupyterLab, Github, and Google Colab.

Figure 2: *JupyterLab logo* [Image from Wikipedia]

Figure 3. *Pytorch, a machine learning framework* [Image from Pytorch.org]

Figure 4. *Github logo* [Image from Github.com]

[1] D. Feil-Seifer and M. J. Matarić, "Defining socially assistive robotics," 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005., 2005, pp. 465-468, doi: 10.1109/ICORR.2005.1501143.

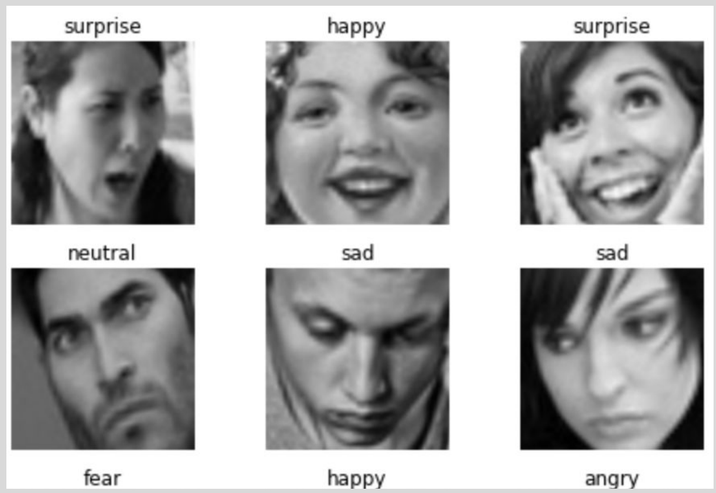

Figure 6. *Part of the data used to train the emotion detection model in Google Colab*
PC: Alondra Cardenas

My STEM Coursework

I was able to relate the work we did in JupyterLab and with Python to my AP Computer Science classes where we learned the basis of computational thinking and coding. I am excited to bring back what I learned this summer working with machine learning and coding to my Advanced Topics in Computer Science class this year. This program will give me a boost in any of my future STEM courses and in my future as I continue on my journey in STEM.

```
def draw_on_frame(frame):
    #make frame gray
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #detect face & draw square around it
    #Detect faces using the haarcascade classifier on the "grayscale image"
    faces = detector.detectMultiScale(gray)

    for face in faces:
        #save the coordinates in x, y, w, d variables
        (x,y,w,d) = face
        # Draw a white coloured rectangle around each face using the face's coordinates
        cv2.rectangle(frame,(x,y),(x+w, y+d),(255, 255, 255), 2)

    #find landmarks on face in frame
    _, landmarks = landmark_detector.fit(gray, faces)

    #draw on the landmarks
    for landmark in landmarks:
        for x,y in landmark[0]:
            #display landmarks w/ white in BGR and thickness 1
            cv2.circle(frame,(int(x), int(y)), 1, (255, 255, 255), 1)

    return frame
```

Figure 5. *A snippet of code from JupyterLab. This function turns each video frame gray to allow the program to detect a face more easily and draw a bounding box around each face as well as find facial landmarks and draw them on the frame before displaying it to the user.*
PC: Alondra Cardenas

Advice for Future SHINE Students / Acknowledgements

My advice to future students is to take advantage of the opportunities offered by SHINE as the 7 weeks go by very quickly. Do not be afraid to communicate and ask for help from your mentors. SHINE is an opportunity to further yourself in STEM, explore your interests, and meet other people who are passionate about STEM and research. It is also okay to use SHINE as an opportunity to figure out what you do and do not like and what careers you may or may not want to pursue. Overall, I would say the most important thing is to take advantage of the opportunities provided, make use of available resources, and have fun!

I would like to thank Dr. Katie Mills for the SHINE opportunity, Professor Maja Matarić for allowing me to participate in the Interaction lab, my SHINE mentor Brandon Thai Tran, my labmate Luke Pratt, my center mentor Cassandra Jeon, and my family for their support.