E.R. – Extra-Terrestrial robot

Carlo ter Woord & Kasper de Vries

Project paper – Robotics 2021
Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
May 2021

Abstract
In this paper we present our findings in creating E.R. the Extra-Terrestrial robot. A study into developing an interactive robotic creature prototype that responds to facial expressions based on the following positive and negative emotions: Angry, Disgusted, Fearful, Happy, Neutral, Sad and Surprised. The process, design, implementation, and results will be discussed.

Introduction
We wanted to explore the possibilities to develop a robotic creature that would be able to respond to emotions from facial expressions using locomotion, our aim was to use both positive and negative emotions.

Our inspiration for the design of the creature comes from the work Senster by Edward Ihnatowicz (Zivanovic (2004)) a large robotic sculpture that interacts with its audience through sound being captured by microphones. Based on the loudness of the audio captured by the robotic sculpture, it moves itself towards the source location with varying speed (e.g. a loud sound: fast and rapid movement, a softer sound: slower movement) (Drucker, 1997). We were interested in using a similar type of interaction within a small robotic creature. We believe that these aspects are interesting to explore as they contain qualities of engagement and can be potentially useful for social Human-Robot interaction.

The challenge of developing such a robotic creature is that it takes a bottom up approach in design, hardware and software architecture and took multiple design cycles as we developed it from scratch. Making it an interesting and novel challenge to explore and come up with a prototype that captures these behavioral qualities and through our aimed functionalities.

Design & Implementation
Designing the robotic creature we started from the ground up with a ‘crab-like’ shape, while using the triangle like shapes inspired by Senster. To explore the basic emotions we created three types of locomotion according to an individual’s facial expression, a forward and backward motion and recurrent motion when in an idle state.

For our hardware we used the Raspberry Pi 4 board for our computation, a Pi Camera to enable the face and emotion recognition, a PCA9685 servo controller board and six TowerPro MG90 micro servos to enable motor controlled movement. Our robotic parts were designed using CAD software (Autodesk Fusion 360) and 3D printed in PLA/PHA plastic.

3D Printed enclosure
Designing our robot using CAD software and 3D printing allowed us to iterate quickly and test the mechanical properties. The design consists of one main body carrying the Raspberry Pi and the servo controller and two arms that each contain 3 servos. Each mechanical part was kept as simple as possible while also being able to add an aesthetic to the designed parts. Our

design was therefore also easy to produce as it only takes about 6 hours to print all necessary components.

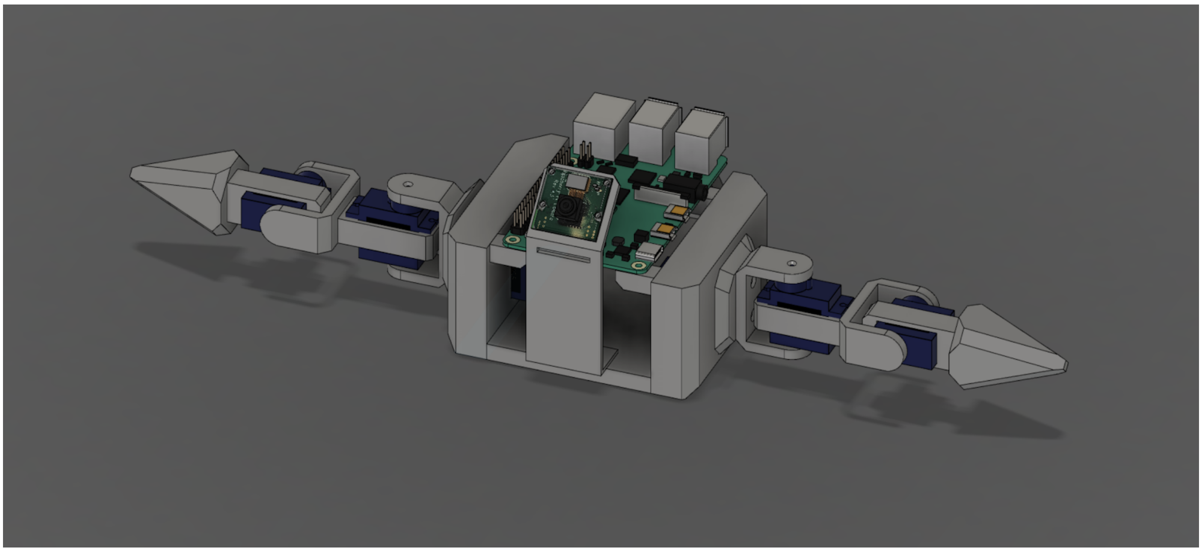

Fig 1. E.R, CAD preview

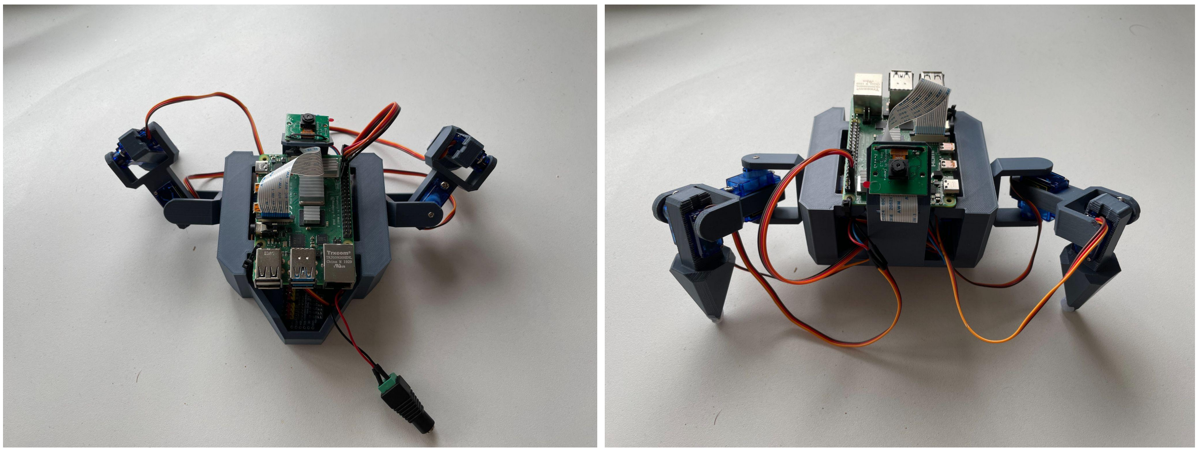

Fig 2. E.R front and back view

Emotion Detection

Using online tutorials we managed to install the OpenCV library on the Raspberry Pi, this required some effort and manual configuration and installation of the dependencies as the current compiler of the Raspberry Pi cannot install OpenCV using a one-line command. After completion, we implemented an emotion detection (including facial recognition) asset which used TensorFlow models to recognise the facial expression of a user. We categorised these emotions into two segments, Positive and Negative expressions.

During the execution of the program, the camera feed is pre-processed to fit the Convolutional Neural Network on which the TensorFlow model was trained. The model would then predict the corresponding emotion each time a face is detected within the camera feed (Fig 4), resulting in the actuation of the robot. These movements are triggered whenever a positive or negative emotion is detected. For positive predictions, the robot would try to approach the person expressing the emotion and whenever a negative emotion is detected the robot would try to flee from the person. If no faces / emotions are detected for an amount of time the robot would initiate it's idle state, a set of random basic movements. After transitioning between states the robot would calibrate the servos negating positioning errors within the servos. Below you can find the visual representation of our program (Fig 3).

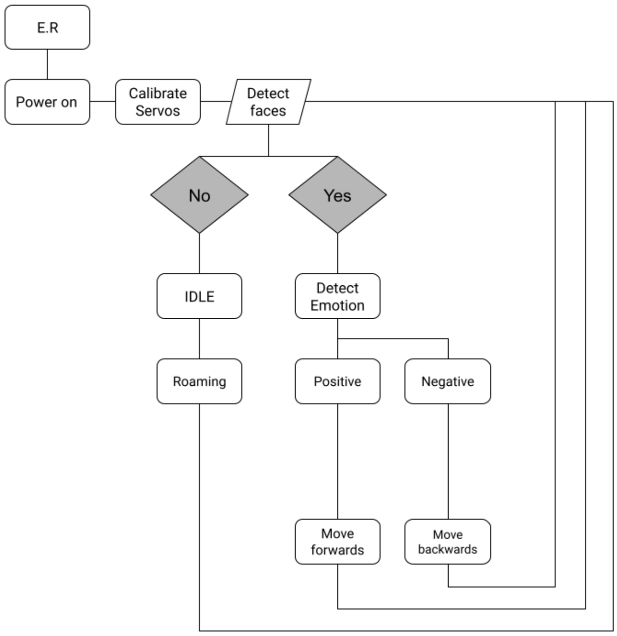

Fig 3. E.R Face state diagram

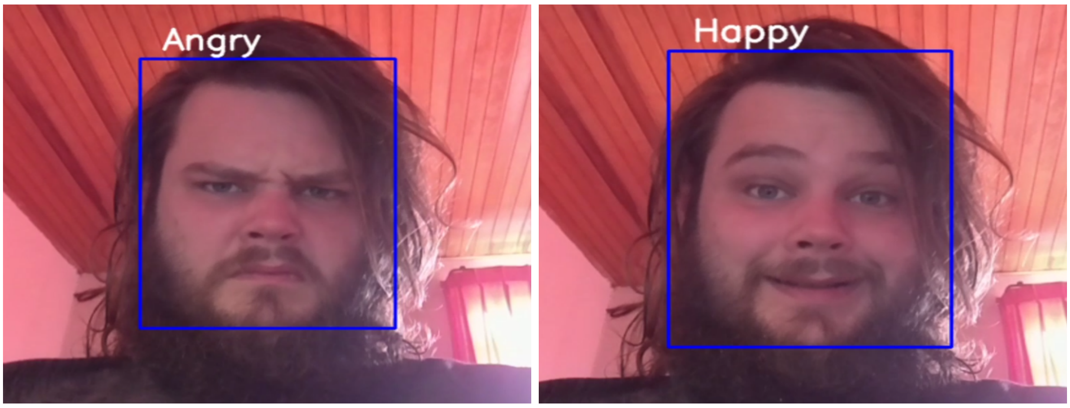

Fig 4. Emotion recognition based on facial expression

Locomotion

Our robot has 3 DOF for each arm this allowed us to develop more extensive locomotion. We developed our motion through an iterative testing process. We used the Adafruit Servo Kit library making it easy to control our servos, as we designed movement through hard coded values (between 0 and 180 degrees), allowing us to have more precise control over how we wanted the leg motion to be controlled. We also experimented with randomised movement for the idle state, resulting in a sense of 'aliveness'. We believe that motion within an idle state is valuable as it provides an affordance to a possible interaction.

Process & Challenges

The main challenge was to get the robot to behave as intended. Getting the motion right to make E.R. move took various iterations to get a working solution. Due to extensive testing, the PLA/PHA parts slowly started slacking, making calibration of the arms sometimes difficult, resulting in unwanted motions at a detected state.

Other challenges regarding the facial recognition aspect of the project was to install the OpenCV library properly, as the compiler of the Raspberry pi cannot simply download and install the library using a single command. Fortunately there are multiple online sources available which explain the process of installation clearly (Shrimali, 2021). Another challenge was to implement the state manager within the program to smoothly transition between different movement states. By keeping our choice of hardware relatively simple, we were able to quickly solve problems if something did not work as intended, making the design process more efficient.

Discussion

Our robot prototype works, however due to the weight distribution our robot does not always perform as optimal. Especially on smooth surfaces our robot is not able to move according to the input. In our tests we used carpets and yoga mats to accommodate this problem and to create more traction.

Interesting to note is that we sometimes got 'free animation' based on the surface it was put on. With 'free animation' we mean behavior and movement not designed beforehand resulting from real world conditions. While designing the robot we did not perform extensive simulations to workout the movements, we tackled our problems in a physical setting, this allowed for good working solutions, however we believe that a simulation could have helped the speed of the development process.

The usage of TensorFlow on a Raspberry Pi was possible due to improved hardware and software. However, we noticed the program expressed a slight delay during the predictions, which could be improved if the used libraries and dependencies were to be fine-tuned to the Raspberry Pi 4. For future developments we would suggest using a different platform like the Jetson Nano, by NVIDIA.

Future research

For future studies it would be interesting to explore inverse kinematics to possibly make the motion design more efficient. Also different arm lengths and body shapes could be explored which might result in different behavior on the given input. Finally, improved human-robot interaction could be explored resulting in additional inputs or movement sets.

References

Drucker, J. (1997). Digital Reflections: The Dialogue of Art and Technology. Art Journal, 56(3), 60–67. doi:10.2307/777824

Zivanovic, A. (2004). Senster. Senster. http://www.senster.com/

Shrimali, V. (2021). Install OpenCV 4 on Raspberry Pi. Retrieved May 14, 2021, from https://learnopencv.com/install-opencv-4-on-raspberry-pi/