Digital Reflections Prototype

November 11th, 2023

Shayne La Rocque / 40182925

Sabine Rosenberg

CART 451

Concordia University

Project Overview

The Digital Reflections project is made with the goal of showing users how they are perceived by AI computer vision models. It takes a photo of the user, captures the perception of this photo several image captioning models, and provides these outputs to use OpenAI's GPT-4 large language model, which then uses them as a basis for generating inferences into the user's identity, and creating a piece of generative art based on their description and those inferences.

The main goal of the project is to present the user with a peak behind the curtain into how they are perceived by various common image-to-text and image classification models. Additionally, adding another layer by creating artwork based on this perception using GPT-4 gives the user some form of return on this transaction, along with the added benefit of supplying visual interest and furthering the discussion about perception and agency in an AI future.

Their input and output data is logged to a database, and after being presented with their artwork and given time to view it, the users are given the opportunity to answer a questionnaire using a generated QR code. This questionnaire response is tied to the input/output data, and analyzed by myself.

Current Project Status

- Under this header, you can describe the current stage of the project, detailing what has been completed and what remains to be done.

Currently the project has the functionality to take a picture of the user, capture the responses of the various image captioners, pass them to the GPT-4 API call, and display the response.

What remains to be done is:

- [ ] Capture user's self-description
- [ ] Logging of input/output to database
- [ ] Creation of unique survey
- [ ] Logging of survey response to database
- [ ] Beautification/Front end

My plans for accomplishing these is:

1. **Implementing User Self-Description Feature:**
    - Two avenues to achieve this:
        - With the release of OpenAI's WhisperV3 and Text to Speech model, I am highly interested in giving a voice to this experience. Rather than simply silently snapping a photo of the user and creating the output, I think it would be very interesting to create a way of collecting the self description and also giving a personality to this experience. It may be hard, and if I get it working I foresee long delays between user and machine in their chatting as we'd be waiting on transcription → Generating response → Generating speech, each time.
        - A simple text entry field.
2. **Establishing Input/Output Database Logging:**
    - Set up a secure and efficient database system for storing all input and output data, including user photos, AI interpretations, and the generated artwork.
3. **Designing and Creating a Unique Survey:**

- Develop a comprehensive questionnaire that effectively gauges user reactions and thoughts about their AI-generated artwork and perceived identity.
    - If I manage to create the text to speech to text idea, I will utilize the same tech for the questionnaire.

4. **Integrating Survey Response Logging:**
    - Link the survey responses to the specific user session, ensuring that each set of feedback corresponds accurately to the relevant input/output data.
5. **Front-End Development and Beautification:**
    - Enhance the user interface and overall aesthetic of the application to make it more engaging and user-friendly.
        - I want to give this experience more life. I have a vision in my mind of an amorphous blob to represent this thing.

- OR, using ChatGPT generated spritesheets to create an animated character that represents the “artist AI”. See below as a quick example, or this Twitter thread where someone found more success.

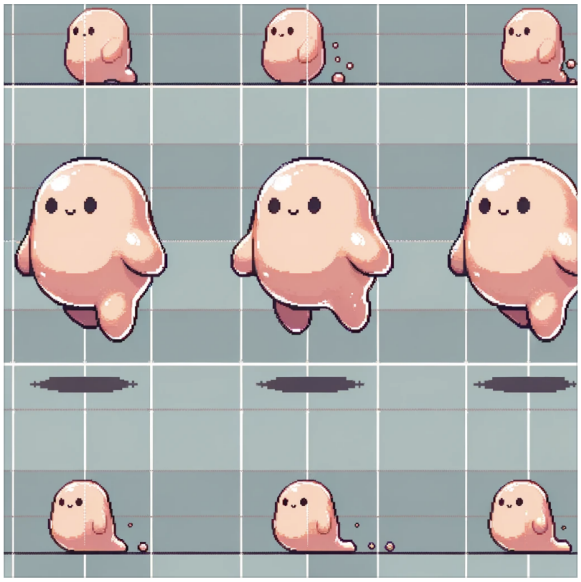
You

Please create a 3×3 spritesheet of an idle animation for a blob, the blob is suspended in the air.

Each piece is a 64×64 grid.

< 2 / 2 >

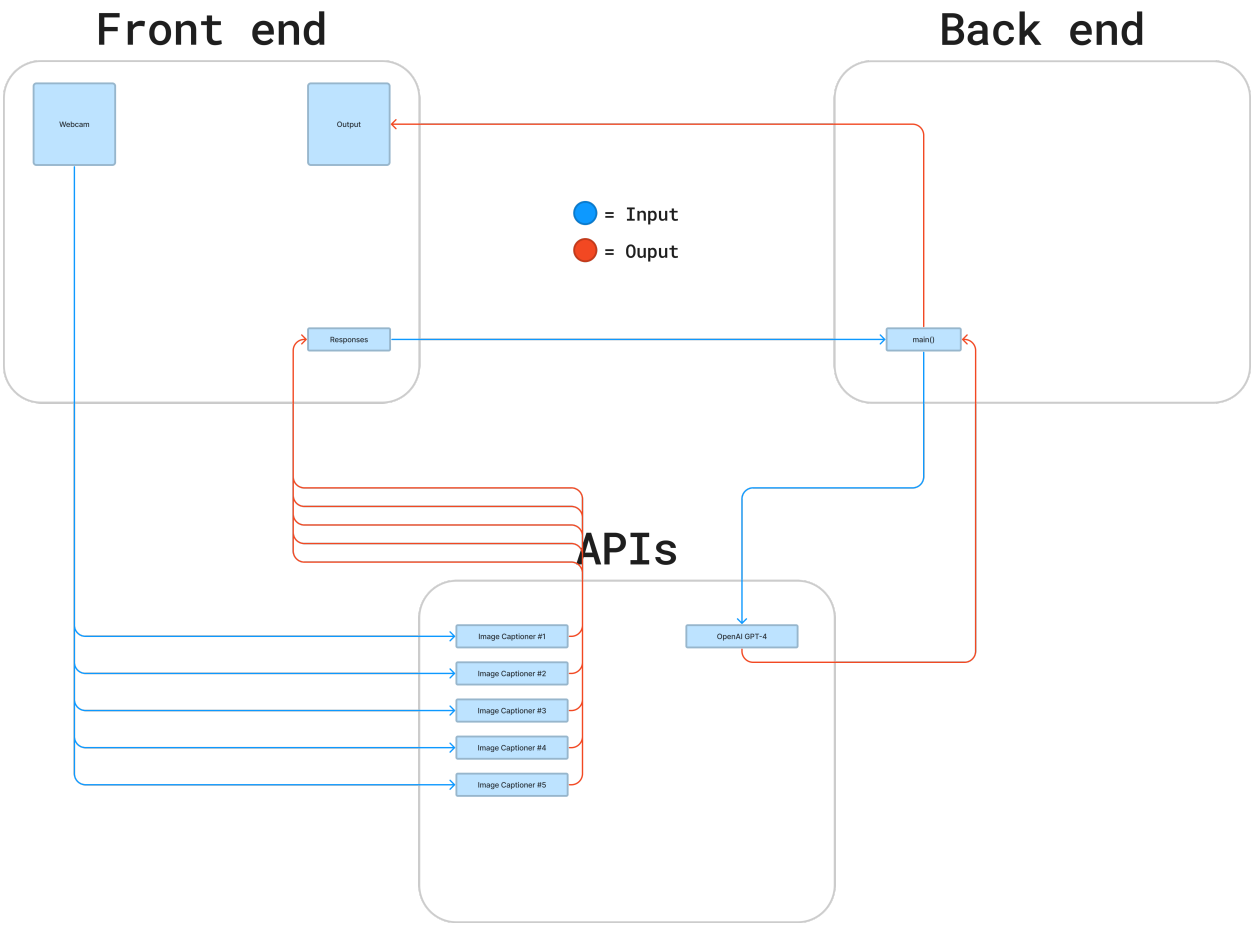
ChatGPT



Here is a 3×3 spritesheet of an idle animation for a blob character, with each piece fitting into a 64×64 grid. The blob is shown suspended in the air, with subtle movements across the frames to depict a gentle floating motion.

System Architecture and Data Flow

As it currently stands, this the architecture:



Component/Feature Analysis

1. Webcam
    a. The webcam component of this app runs on the front end. It captures a picture of the user, blobbifies it, then sends it off to various Huggingface Inference API models. Upon receiving a response, it automatically inputs it into respective form fields.
2. Responses
    a. The responses form is where the output from Huggingface is captured. This collects the user input from the form and sends it to the server as a JSON object. The server then uses the OpenAI API to generate a JSON response containing inferences and P5JS code based on the user's descriptions.
3. Main/endpoint

a. This is a Node.js server-side JavaScript code with an Express.js web server. It listens on port 3000 and has a single endpoint /generate-art that accepts a POST request. The endpoint triggers an OpenAI API call to generate a JSON response containing inferences and P5JS code based on the user's descriptions. The main function takes in six user descriptions and a self-description, and uses them to generate a chat completion using the OpenAI API.

Project Feature Assessment

- All currently implemented features are working fairly close to the goal, with the aside that GPT-4 does not always try to force a guess about the users profile if not supplied with relevant data (Most notable, ethnicity).
- Additionally, it seems to consider each image description as a description of a unique image, rather than several different descriptions of the same image. This makes the output have a different tone, for example in takeaways it will often say: “The user is often seen working with a laptop” if each image descriptions mentions a laptop. This causes skewed art generation as it will put more weight into these elements than it potentially would if it considered each description as a different description of the same image. To fix this I need to tweak the prompt to let the model know that these descriptions are all by different computer vision models of the same image.