Part 1 – Planning:

Problem

In short, we would like to enable chat-like interactions between users and our multimodal model. This would improve our application by enhancing user experience, by leveraging past interactions to provide more accurate and targeted feedback on the surroundings of the user. In addition to the real-time AI image captioning service, we would require AI-driven speech-to-text and text-to-speech functionalities, along with more “general intelligence” to properly interpret a multitude of user queries. The ultimate goal is to allow interactions with our application to be indiscernible from seeking assistance from a human companion.

For example, a simple user story would be a visually-impaired user asking for clarification or additional detail from a previous response. The current iteration of our application only allows users to make a single API call with a simple query to describe the provided image, which does not take into account past context. Thus, each interaction between the user and our application is currently completely isolated. However, we believe that a sense of temporal coherence would allow our application to provide more informed responses, and better cater to the specific needs of users. A user might ask: “describe the painting in front of me”, and our app might initially respond with: “The image shows a serene landscape at sunset. In the foreground, there's a field of wildflowers with a mix of violet and yellow blooms. Beyond the flowers, there are gently rolling green hills and fields, with patches of darker green forests…”; but then, the user might cut them off and ask: “great, but how many trees are in the forest?”; instead of mindlessly rambling on, or providing another general description. or requiring clarification on what “forest” we are referring to, we might expect that the application to answer “there are 31 conifers in the painting” like a human might.

Criteria, Research, Selection

Our criteria for success can be summarised as follows:

- Leverage past interaction context to provide more insightful responses.
- Allow for mid-response “cut-off” which is followed by a quick “reaction”.

To fulfill these criteria, we could consider fully integrating OpenAI’s Chat Completion API contextual based responses. Moreover, we would need to creatively incorporate multimodal data in the form of images into this contextual pipeline. Hence, we need to identify models that are multimodal and are capable of making use of a context window that involves images. OpenAI’s 4o offering seems to fit this use case quite well as it allows for messages of type “text” and “image” to be passed as context to the model. This solution would require some extra work on our part to properly preprocess the images, ensuring they are in a format suitable for the model. Additionally, this would potentially require us to store past “conversations” in order to build a web of crucial information that could boost our ability to be our user’s eyes, potentially allowing us to gain useful insights about the user that would make interactions more natural.

Despite our preference for OpenAI, there are a wide variety of open source multimodal models that would be able to achieve similar performance to OpenAI's offerings at a potentially lower cost as well. This would include models like the Llama 3.2 from Meta, or the Qwen2-VL from Alibaba, which enjoy much smaller parameter sizes. Various LLM inference providers support the use of these models such as Together AI, along with simple APIs that would work well with our application. The main downside is that we have already worked with and integrated OpenAI's stack into our existing application and our main goal would be to prove out a working prototype of these features above all else. Tweaking parameters like which model we are using will be important later but not a pressing consideration for an MVP.

As justified in the "problem" and "impact analysis" sections, these new changes would greatly improve our AI-driven responses; nevertheless, these changes would also introduce new ethical problems that need to be considered. While storing user data, especially conversation history, we must be mindful of protecting user privacy. Overall, this can be addressed by providing clear terms-and-services conditions, and storing as little user data as possible for as short a time-frame as possible. This reduces user exposure to harmful data leaks, and could make our application more approachable to users. Moreover, we could potentially consider allowing users to fully "opt-out" of these history features.

Part 2 – Impact Analysis:

As we mentioned before, we anticipate a smoother user experience that is more personalized and natural. Most of all, we hope to enhance the quality of description produced by our application which would be our main goal with this proposal. One way we could test the delta between our baseline would be to construct a testing set with a series of images taken in succession. Then after asking the models to provide descriptions and responses to a set of follow up queries on these test set examples, a human annotator could then inspect the responses and score it based on some reasonable set of criteria (accuracy, helpfulness, succinctness, etc). This way we can reliably measure the relative improvement compared to single isolated descriptions. We could potentially automate some testing by having some "expected" responses which could be verified in unit-tests.

In terms of trade-offs, a known risk for our application in general is the overall latency and responsiveness to user requests due to multimodal response generation with a mix of images, text and data. The round trip time from our device to OpenAI could be quite noticeable. Anecdotally, even OpenAI's ChatGPT app hangs and struggles to respond quickly under idealized conditions (i.e having a high speed WIFI connection, no other additional app overhead like with our camera). This would be one glaring reason to consider going with smaller models as this could ease some latency of our application's critical path.