LookoutX: Smart Assistant for the Visually Impaired Project Midway Report

Shreayan Chaudhary, Hasan Muhammad

Motivation

People with visual impairments or disabilities face significant challenges in their daily lives, especially when it comes to navigating unfamiliar environments or accessing information. Smart assistants can provide a valuable tool to assist these individuals in their day-to-day lives. By leveraging cutting-edge computer vision and natural language processing technologies, we can build a smart assistant that can interpret the user's surroundings and answer their questions in real-time, making it easier for them to navigate the world around them. Questions can be general or based on the environment like “Is the traffic light red?”, “Is the coffee store open?”, “How crowded is the Post Office?”, etc. Also, the applications of this system are limitless.

Hypothesis

We hypothesize that by combining state-of-the-art Computer Vision and Natural Language Processing techniques, we can develop a smart assistant that can answer a wide range of questions asked by visually impaired or disabled individuals based on real-time video feed captured using a camera. Furthermore, we expect that integrating the system with other modules such as GPS, Google Maps, and other APIs will seamlessly increase its convenience and usability for users.

Experiments

We propose to conduct the following experiments to evaluate the performance of our system:

Experiment 1: Question Answering Accuracy: We will evaluate the accuracy of the question-answering module of our system by comparing its responses to a set of human-generated answers for a variety of questions, including both general knowledge questions and questions specific to the user's surroundings. We will use standard evaluation metrics such as accuracy, precision, recall, and F1 score to measure the performance of our system.

Experiment 2: Real-time human evaluation: We are going to create a couple of (let’s say 30 question and image pairs) and manually evaluate the performance of the model on those samples. This will function as a small dev set to ensure that our model can handle the use cases we are actually interested in.

Experiment 3: Integration with APIs: We will evaluate the effectiveness of integrating our system with commonly available APIs and services like GPS, Google Maps and Places APIs, Folium, Nominatim, etc. We will test if our system is correctly identifying the user's location and answering based on the location, if relevant. To the best of our knowledge, there are no datasets we could use for automatic evaluation, so this will have to be done manually.

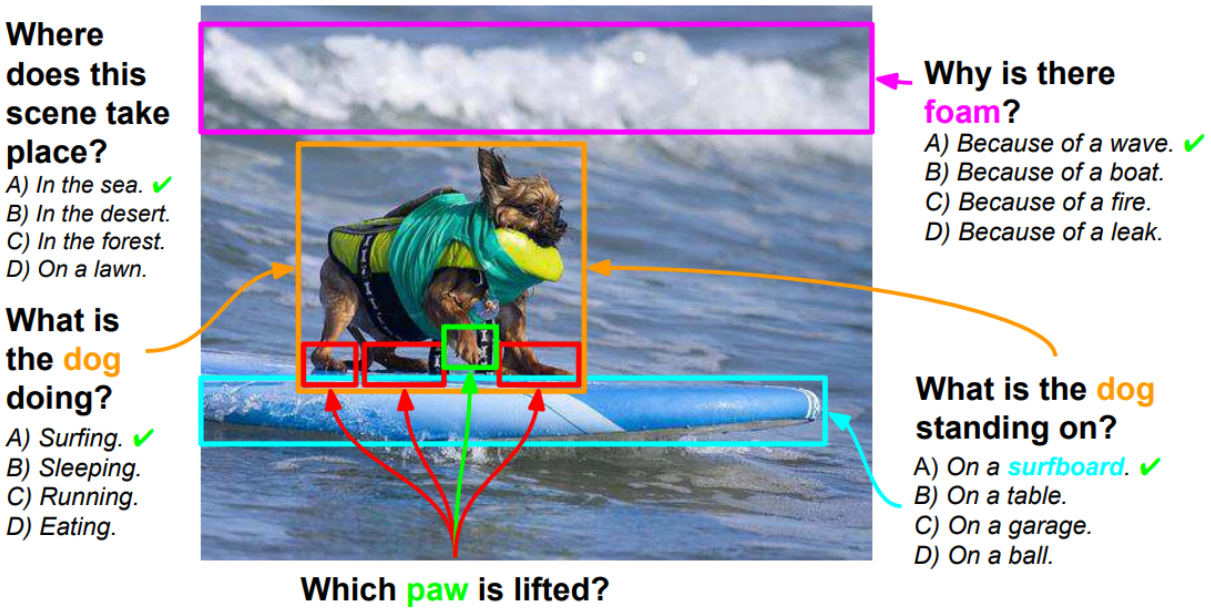
Datasets

We will use the following Visual Question Answering datasets for training and evaluating our system:

COCO: The Microsoft Common Objects in Context (COCO) dataset will be used to train and evaluate our computer vision models. COCO consists of over 330,000 images with more than 2.5 million object instances labeled with bounding boxes. Note: This is not used directly since VisualQA (defined below) has more relevant data that draws from COCO.

VisualQA data: VQA has ~205k images from COCO, 1M questions, and 11M ground truth answers.

Visual7W: Visual7W is a dataset of images with lots of questions that amount to a very dense annotation of the image.



Baseline Model

We are using the vision-language model OpenFlamingo by Awadalla, et al. (2023) out of the box.

Methodology

We propose to use the following methods to build our system:

1. Use DroidCam as a camera that will record the environment via a live video feed. Later, it can be replaced by an embedded camera in smart glasses. It records the videos and sends them to the system.
2. A speech2text module will be used on the audio spoken by the user to convert the user's query from audio to natural language - let's say we call the text the question.
3. As soon as speech2text activates, we will capture a few frames at regular intervals (frequency to be experimented with - let's say 5 frames every second) and save these images.
4. Feed the image and question to ChatGPT 4 API if available (most likely it will not be made available). If ChatGPT 4 API is not available, then we will use the BEiT model as mentioned above. Time permitting, we will look into training our own model that will encode the image using CLIP. CLIP (Contrastive Learning Image Pretraining) is a state-of-the-art model that can simultaneously perform image and text classification. Sheng S et al. (2021) shows that CLIP can be used as an image encoder in QA contexts, which is capable of generating high-quality answers to a wide range of questions.
5. During the eval phase, get the model answer and decode it to get the answer to the question.
6. We will integrate our system with common APIs like GPS and Google Maps to provide additional information based on the user's surroundings. This will help with questions like "How long will it take me to walk to the nearest medical store?", "Is there traffic on the Saint Paul Street?"

Halfway Progress and Preliminary Results

1. We have successfully set up the video live stream capture module. We have used the tool called DroidCam to use the phone as a stub or proxy, and it can be replaced by an embedded camera in smart glasses later. It records the videos and sends them to the system. The image shows how it is being used currently:

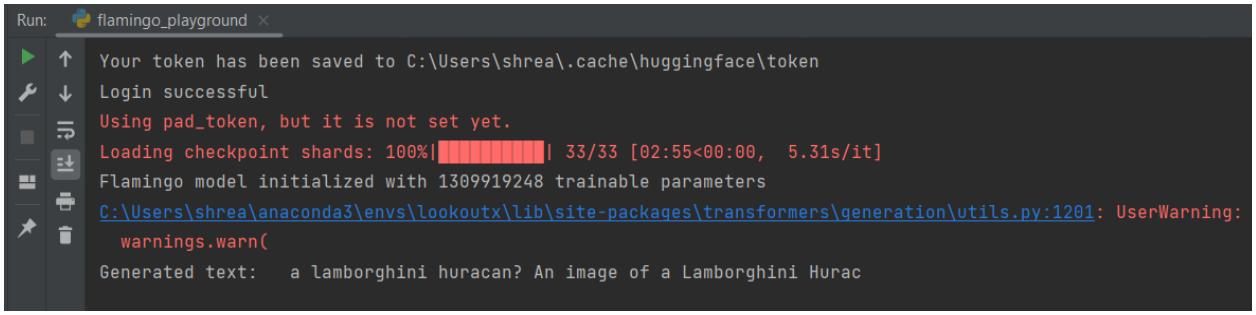
2. We have set up the OpenFlamingo (as given in the project proposal feedback) for the Vision-Language model. Since it is a Vision Language transformer, there are 2 components to it. We have used CLIP ViT-L/14 and LLaMA-7B, which are the smallest CLIP ViT and LLaMA models respectively, due to a lack of computation resources for larger models. We have written the image preprocessing and the text preprocessing modules that allow us to feed real images and natural text into the model to generate real output.
   Image transformations: Image format conversion -> Resizing -> Convert to (N, C, H, W)
   Text transformations: Tagging -> Tokenization -> Encoding -> Decoding generated output

   We have tried playing around with the model and done a couple of inferences on new images and text. We tried both 0-shot and few-shot prompts to the model to see how well it does out of the box. The model outputs are given below:

Test Image:

Prompt / Question: (2-shot on images of cat with the prompt “<image> this is a cat <endofchunk> <image> this is a bathroom sink <endofchunk>”)
what is this thing?

Output: a lamborghini huracan? An image of a Lamborghini Huracan



Test Image 2:

Prompt / Question: (0-shot) what does this image describe?

Output: a group of penguins? An image of a group of penguins.

```
flamingo_playground
Your token has been saved to C:\Users\shrea\.cache\huggingface\token
Login successful
Using pad_token, but it is not set yet.
Loading checkpoint shards: 100%|██████████| 33/33 [03:19<00:00,  6.06s/it]
Flamingo model initialized with 1309919248 trainable parameters
C:\Users\shrea\anaconda3\envs\lookoutx\lib\site-packages\transformers\generation\utils.py:1201: UserWarning: \
  warnings.warn(
Generated text:   a group of penguins? An image of a group of penguins. An image of
```

Test Image 3:

Prompt / Question: (0-shot) is the coffee shop open?

Output: Starbucks Coffee - 10000 W 75th St

```
flamingo_playground
Your token has been saved to C:\Users\shrea\.cache\huggingface\token
Login successful
Using pad_token, but it is not set yet.
Loading checkpoint shards: 100%|██████████| 33/33 [02:39<00:00,  4.85s/it]
Flamingo model initialized with 1309919248 trainable parameters
C:\Users\shrea\anaconda3\envs\lookoutx\lib\site-packages\transformers\generation\utils.py:1201: UserWarning:
  warnings.warn(
Generated text:   | Starbucks Coffee - 10000 W 75th St
```

3. We have also set up the audio module to record speech prompts. We are figuring out how to clip and extract the meaningful portion of the audio stream. For the ASR on this audio file, we are using the OpenAI Whisper by Radford, et al. (2022) model, which is a state-of-the-art open-source model for audio recognition tasks. We have set up the Whisper module, and are trying to crop parts out from the audio stream to send to the Whisper module.

Challenges faced:

1. The OpenFlamingo model has 13B trainable parameters. As a result, it takes around 6-7 minutes on the local system (CPU: AMD Ryzen 9 5000, GPU: RTX 3050Ti 4 GB) to load the downloaded model weights. Since it is a Vision Language transformer, there are 2 components to it - CLIP ViT-L/14 and LLaMA-7B. As a result, loading the weights for both models take a while.
2. While we are able to load the model in the memory and run it successfully on 0-shot and few-shot prompts, the inference time for new inputs is quite significant. It takes around 5-8 minutes for the model to run inference on new image-text example pairs.
3. While we are using the OpenAI Whisper model for ASR, extracting meaningful parts from an audio stream is very challenging. There are three primary approaches used for this task: One approach is to try to crop parts out from the audio stream, and another approach is to use a button that is clicked whenever the user wants to speak a prompt. Currently, for devices like Google Home, Alexa, etc., they have integrated a mini model in the device hardware which will actively listen only to prompts like “Hey Google” and “Hey Alexa”, and once these words are detected, it will activate the larger ASR model to capture the prompt from the user. We are figuring out how to clip and extract the meaningful portion of the audio stream.

Next Milestones

1. We plan to evaluate the results of the OpenFlamingo model on the VQA dataset.
2. We plan to create a dataset with real images and their corresponding ground truth answers and evaluate the results on it to evaluate the performance of the system empirically in real-life use cases.
3. Time permitting, we plan to integrate our system with common APIs like GPS and Google Maps to provide additional information based on the user's surroundings. This will help with questions like “How long will it take me to walk to the nearest medical store?”, “Is there traffic on Saint Paul Street?”

Extensions and Future Scope

Given that we implement the proposed system mentioned above, here are some possible experiments/extensions that we might work on, time permitting:
Experiment 3: Provenance/Localization: We will evaluate the effectiveness of our system in providing provenance or localization information for the answers generated. We will use a set of annotated images to test if our system is correctly identifying the regions of the image that the answer corresponds to.

Expected Outcome

We expect that our system will be able to accurately answer a wide range of questions asked by visually impaired or disabled individuals based on real-time video feeds captured using a

camera. We also expect that integrating the system with other modules such as GPS, Google Maps, and other APIs will increase its convenience and usability. We hope to improve the daily lives of blind and disabled individuals by making some tasks and activities convenient for them.

References

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). Vqa: Visual question answering. In Proceedings of the IEEE international conference on computer vision (pp. 2425-2433).

Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K. W., ... & Keutzer, K. (2021). How much can clip benefit vision-and-language tasks?. arXiv preprint arXiv:2107.06383.

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., ... & Wei, F. (2022). Image as a foreign language: Beit pretraining for all vision and vision-language tasks. arXiv preprint arXiv:2208.10442.

Zhu, Y., Groth, O., Bernstein, M., & Fei-Fei, L. (2016). Visual7w: Grounded question answering in images. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4995-5004).

Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems, 35, 23716-23736.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv:2212.04356.

Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems, 35, 23716-23736.