

ETY-Translator

ECE: 5995 General AI Tools (GAIT) Final Project

Group 6: Yaohua Wang, Yijie Zhang, Yan Chen

February 20, 2024

Abstract

When a foreigner lands in a foreign country, communication becomes an urgent challenge. Not only does the foreigner need to communicate with local people vocally, but he also needs to read foreign text messages to order food or to read signs. Translating devices nowadays need to identify the language to translate and require users to input the text message. Such a method is reliable but takes a long time to exchange the message, causing asynchronous communication. Applications that translate content from an image also exist but only translate the embedded message itself. Users cannot ask follow-up questions about the translated content from the image. To solve such shortages, we proposed the ETY-Translator that smashes the language barrier by using speech-to-speech translations with auto-language detection and extracting useful information from the user-captured images with chatting capabilities that allow users to ask for details in the image or simply chat about anything, thanks to the OpenAI APIs.

1 Introduction

Speech translators are vital for global communication and breaking down language barriers in various contexts. In business, they facilitate international trade and negotiations, while in travel, they assist tourists in navigating unfamiliar environments. They contribute to inclusivity by making events and activities accessible to individuals with different linguistic abilities. Advances in technology continue to improve the accuracy of real-time speech translation, making it an indispensable tool in our interconnected world. However, most current translators provide only text-to-text translation with an instruction from the user indicating the source and target language. It fails to provide a translation service when the user is unaware of the target language or struggling to type the source language, which is often the case when traveling abroad.

To improve translator accessibility with the help of generative AI tools, we developed a smart two-way speech-to-speech translator to automatically detect and translate speech between two different

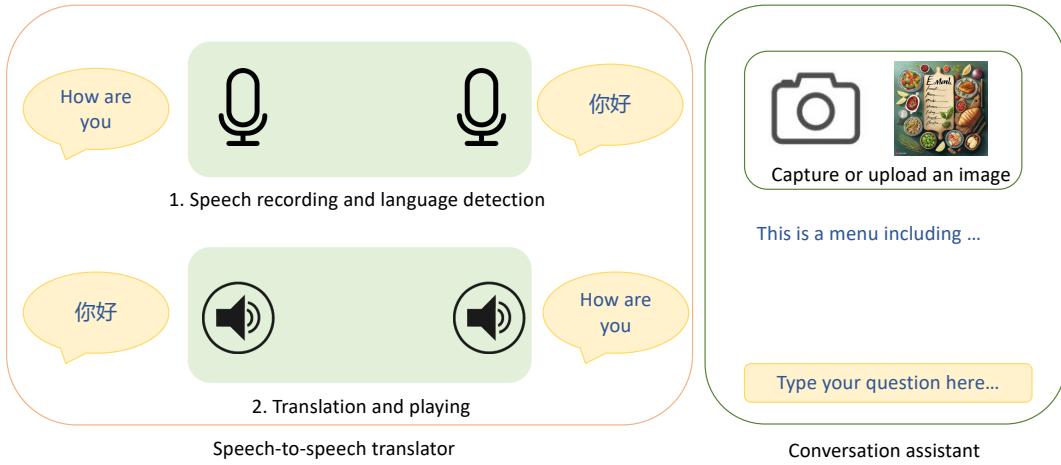


Figure 1: Pipeline of ETY-Translator

languages. The smart translator offers a platform on which users can chat interactively without language gaps. In addition to automatic speech-to-speech translation, we incorporate a conversation assistant to answer any questions the user may have. Translation is the first key step in communicating with people who speak different languages. The culture gap is more difficult to address in the short term. An additional helper with comprehensive knowledge would be helpful in tackling the culture gap and improving communication efficiency. The flexibility to capture or upload an image increases the capacity of the conversation assistant.

The comprehensive smart translator, namely ETY-translator, aims to provide automatic speech translation combined with a conversation assistant to help users break the language barrier.

2 Pipeline of ETY-Translator

The main features of the ETY-Translator are two-fold. The first part is the speech-to-speech translation, which involves transforming speech into text, detecting the environmental language, translating the text into the environmental language, and transforming the translated text back to the speech. The second part is the image-based conversation assistant where users can either capture a photo with webcams or upload an image from the local memory. Once the image is uploaded, a summary will be replied to a chat box where users can ask follow-up questions about the content in the image or ask anything they want. The translator pipeline is shown in Fig.1.

3 Generative AI Tools and Implementation Details

We incorporate multiple generative AI tools to implement the proposed ETY-translator. At the starting stage of this project around November 6th, 2023, OpenAI had its first Development Day and

announced its new environment upgrade including customized GPT models, DaLLE-3 APIs, and a new text-to-speech model. With so many powerful tools and APIs OpenAI announced, we decided to all in OpenAI by building this project with only OpenAI APIs.

Specifically, we generated our group icon image using DaLLE 3; used Python with Flask library to build the server; used Python to handle all the OpenAI's APIs; used JavaScript to build the connection between the front end and back end; and finally used HTML with Bootstrap to control the layout of the front end web page. We introduce the implementation details of each AI tool separately.

3.1 Speech-to-Speech Translation

In this section, we cover the technical aspects of the speech-to-speech translation. The speech-to-speech translation can be divided into two parts:

1. speech-to-text (STT) and text-to-speech (TTS).
2. language detection and translation.

3.1.1 Speech-to-Text (STT) and Text-to-Speech (TTS)

The idea for the speech-to-speech translation starts with using Whisper, OpenAI's STT model, to transcribe whatever the user speaks into the text message in the user's language, i.e., if the user speaks English, Whisper transcribes it into English content; if the user speaks Spanish, it transcribes it into Spanish content. Since the transcribed content is text messages, we directly returned the jsonified message to the front end.

To transform the text message back to audio, we used the TTS API from OpenAI where the text message is transformed back to audio in the same language, i.e., written Spanish is transformed into Spanish audio. Since the transformed audio is not text messages anymore, we returned it as an audio blob to the front end.

3.1.2 Language Detection and Translation

Now that we can do STT and TTS, the remaining question is how do we translate the text messages. At first, we thought to build a drop-down menu to give users a variety of choices but we later found out this requires us to write all types of languages in their same types, i.e., writing "Spanish" in Spanish, "Chinese" in Chinese, etc. Otherwise, the user cannot recognize which language they are speaking. Moreover, even if the user recognizes their home language, they don't know what language other people are speaking. Therefore, they need to hand over the device to the local people and ask them to choose from the menu to find their local language.

Speech-to-speech Translation

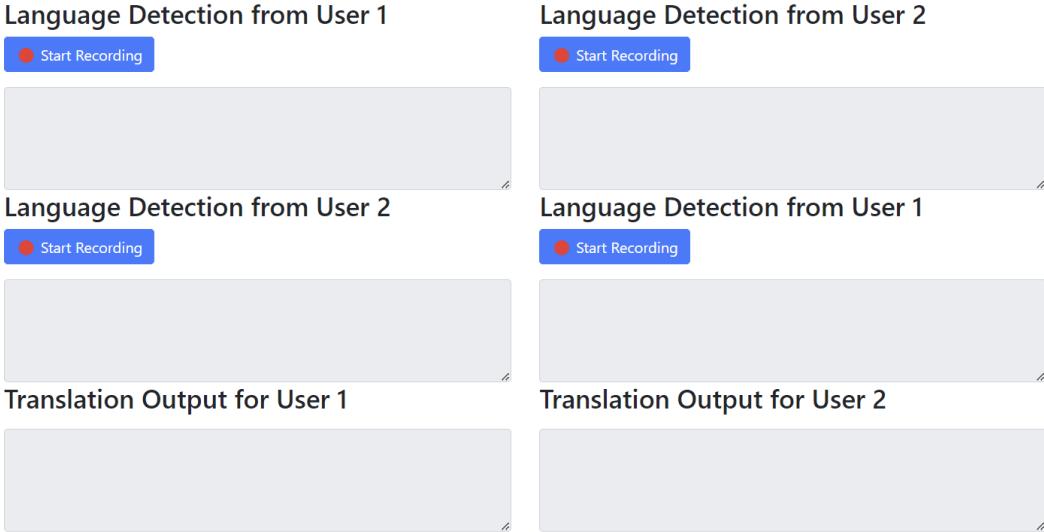


Figure 2: Layout of the Speech-to-Speech section in the web page

As a result, we decided to build two ChatGPTs to handle the issues. The first GPT was built to be a language detection assistant to detect what kind of language occurred in the surroundings. This GPT is prompted to only return the type of the language in English without saying any other words. Then the second GPT, which was built to be a translation assistant takes the transcribed text and translates it into the language type the first GPT detected. In such a way, the user doesn't need to identify the language he wants to translate himself, instead, surrounding languages are automatically detected, and transcribed text messages are translated.

The final layout of the deliverable is shown in Fig. 2. The left column is for user 1 and the right column is for user 2. After user 1 records what he wants to say using the left-top recording button, the left-middle recording button is used to detect the surrounding languages. Once the language type is detected, the translation is activated and the translated audio is automatically played after it's ready. User 2 uses the right column and follows the same steps to get the translated audio played. Once the language type is defined, users no longer need to detect the surrounding languages every time they want to say something. The translated audio is fixed to the detected language unless a new recording to detect the languages.

3.2 Image-based Conversation Assistant

In this section, we cover the technical aspects of the Image-based Conversation. The Image-based Conversation can be divided into two parts:

1. Image analysis.

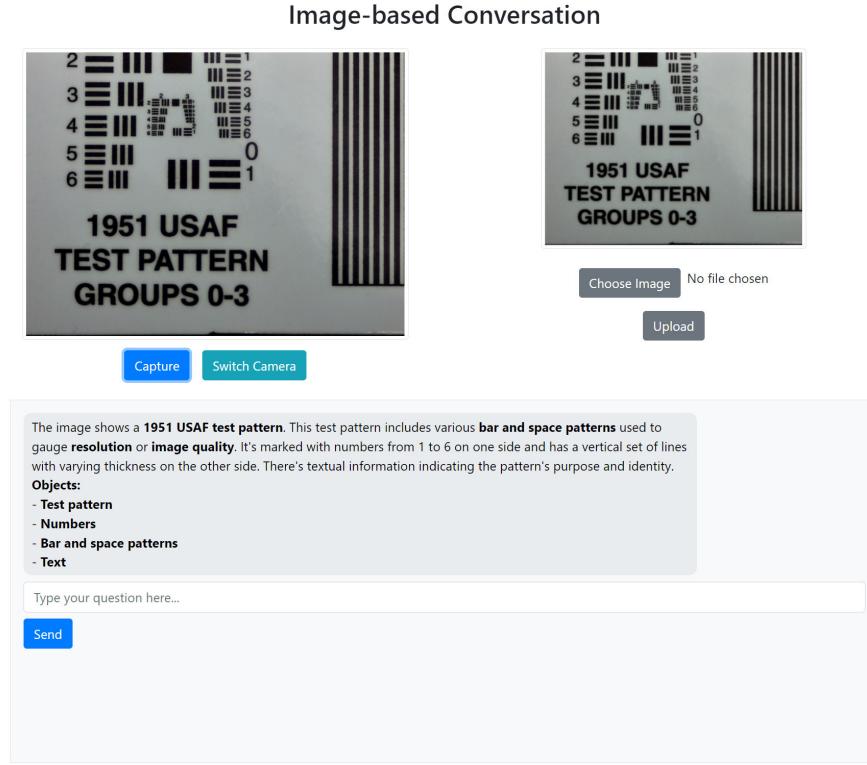


Figure 3: User Interface of Image-based Conversation

2. Conversation assistant.

3.2.1 Image analysis

In this part, we exploit OpenAI’s “gpt-4-vision-preview” model to process the analysis of the submitted image. There are two options to submit the image. As shown in Fig.3, the first option is to ”Capture” an image of the real-time camera, and the second option is to ”Choose Image” file and then ”Upload” it. Submitted pictures can be previewed in the upper right corner of the interface.

In the initial state of submitting an image, a built-in prompt, ”What’s in this image? Be descriptive...” will be added to generate a simple and relatively comprehensive description of the image. Here the ChatGPT API will call a function with only image parameters.

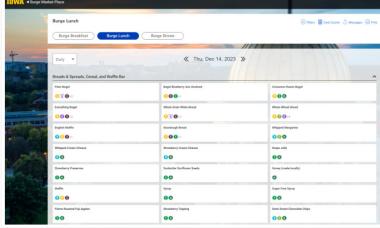
3.2.2 Conversation assistant

In this part, we continue to exploit OpenAI’s “gpt-4-vision-preview” model to process the analysis of the submitted image plus the user message. After the image is submitted, as shown in Fig.4, the user can communicate with the AI assistant, for example, asking questions about the menu: Could you list the food in the image in order? And the AI assistant offers the answer correctly. Here the ChatGPT

Image-based Conversation



Capture **Switch Camera**



Choose Image **BurgeLunch_Breads & Spreads.jpg**

Upload

The image displays a screenshot of a website for the **Burge Market Place** at the University of Iowa, focusing on the menu for **Burge Lunch**. The menu is categorized under **Breads & Spreads**, **Cereal**, and **Waffle Bar**. Across the top, there are tabs for **Burge Breakfast**, **Burge Lunch**, and **Burge Dinner**, indicating different meal options. There is also a date selector showing **Thu, Dec 14, 2023**. Below the header, the menu items are listed with accompanying icons indicating various dietary preferences or restrictions such as gluten-free, vegetarian, vegan, and whether items contain nuts or eggs. For example, **Plain Bagel** and **Everything Bagel** each have certain icons next to them, including a thumbs-up icon probably signifying popularity or recommendation, and a numerical indicator with a plus sign suggesting more information or options. Other items listed include **English Muffin**, **Whipped Cream Cheese**, **Strawberry Preserves**, **Waffle**, and others, along with their respective icons. Each item has a separate white bar, making it easy to distinguish between different options. The background of the menu looks to be translucent, overlaying what seems to be an image of

Could you list the food in the image in order?

The image shows a list of foods available under the category "Breads & Spreads, Cereal, and Waffle Bar" for a meal labeled "Burge Lunch." The food items listed are as follows: 1. Plain Bagel 2. Everything Bagel 3. English Muffin 4. Whipped Cream Cheese 5. Strawberry Preserves 6. Waffle 7. Flame Roasted Fuji Apples 8. Bagel Blueberry 4oz Unsliced 9. Whole Grain White Bread 10. Sourdough Bread 11. Strawberry Cream Cheese 12. Sunbutter Sunflower Seeds 13. Syrup 14. Strawberry Topping 15. Cinnamon Raisin Bagel 16. Whole Wheat Bread 17. Whipped Margarine 18. Grape Jelly 19. Honey (made locally) 20. Sugar-Free Syrup 21. Semi-Sweet Chocolate Chips These items are listed on a food service website or app interface with interface elements such as filters, a calendar view, and allergen information represented by icons.

Type your question here...

Send

Webcam is local on your browser, and **no images or keys are saved** by this demo.

Dev'd by [G6 of ECE5995 GAIT \(UIowa\)](#)

Figure 4: Image-based Conversation Chatting with Image

API will call a function with image parameter and user message.

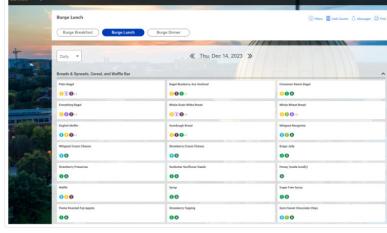
As stated in our main motivation, the main purpose is to help the communication of information for foreigners. So, for example, for this English menu, parents from China may not speak much English. They can ask in their native language (Chinese): Are there any foods in this menu screenshot? Can you please translate it into Chinese (in Chinese)? Then, as shown in Fig.5, they got the expected menu in Chinese and English.

Finally, if the user needs to discuss or ask questions that are not related to the picture, the relevance of the question will be automatically identified. If it is not related to the picture, then a regular discussion or question and answer can be performed, as shown in Fig.6. Here the ChatGPT API will call a function with only the user message.

Image-based Conversation



Capture
Switch Camera



Choose Image
BurgeLunch_Breads & Spreads.jpg

Upload

the top, there are tabs for **Burge Breakfast**, **Burge Lunch**, and **Burge Dinner**, indicating different meal options. There is also a date selector showing **Thu, Dec 14, 2023**. Below the header, the menu items are listed with accompanying icons indicating various dietary preferences or restrictions such as gluten-free, vegetarian, vegan, and whether items contain nuts or eggs. For example, **Plain Bagel** and **Everything Bagel** each have certain icons next to them, including a thumbs-up icon probably signifying popularity or recommendation, and a numerical indicator with a plus sign suggesting more information or options. Other items listed include **English Muffin**, **Whipped Cream Cheese**, **Strawberry Preserves**, **Waffle**, and others, along with their respective icons. Each item has a separate white bar, making it easy to distinguish between different options. The background of the menu looks to be translucent, overlaying what seems to be an image of

Could you list the food in the image in order?

The image shows a list of foods available under the category "Breads & Spreads, Cereal, and Waffle Bar" for a meal labeled "Burge Lunch." The food items listed are as follows: 1. Plain Bagel 2. Everything Bagel 3. English Muffin 4. Whipped Cream Cheese 5. Strawberry Preserves 6. Waffle 7. Flame Roasted Fuji Apples 8. Bagel Blueberry 4oz Unsliced 9. Whole Grain White Bread 10. Sourdough Bread 11. Strawberry Cream Cheese 12. Sunbutter Sunflower Seeds 13. Syrup 14. Strawberry Topping 15. Cinnamon Raisin Bagel 16. Whole Wheat Bread 17. Whipped Margarine 18. Grape Jelly 19. Honey (made locally) 20. Sugar-Free Syrup 21. Semi-Sweet Chocolate Chips These items are listed on a food service website or app interface with interface elements such as filters, a calendar view, and allergen information represented by icons.

这个菜单截图里有什么食物吗？请翻译一下中文，谢谢！

这张图片显示的是一份菜单截图，菜单来自于“Burge Market Place”，并且着重显示的是“Burge Lunch”（Burge 午餐）的一部分。菜单上列出了各种面包和涂抹品，早餐谷物和华夫饼吧台的选择。以下是食物和一些配料的翻译：
- Plain Bagel - 素百吉饼 - Everything Bagel - 杂粮百吉饼 - English Muffin - 英式松饼 - Whipped Cream Cheese - 打发奶油芝士 - Strawberry Preserves - 草莓果酱 - Waffle - 华夫饼 - Flame Roasted Fuji Apples - 火焰烤富士苹果 - Bagel Blueberry 4oz Unsliced - 4盎司整粒蓝莓百吉饼 - Whole Grain White Bread - 全谷物白面包 - Sourdough Bread - 酸面包 - Strawberry Cream Cheese - 草莓奶油芝士

Type your question here...

Send

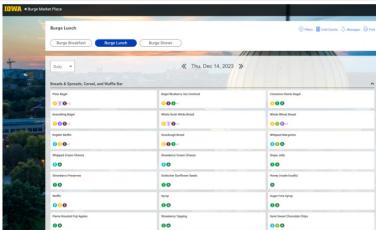
Webcam is local on your browser, and **no images or keys are saved** by this demo.
 Dev'd by [G6 of ECE5995 GAIT \(U Iowa\)](#)

Figure 5: Image-based Conversation Chatting in Foreign Language

Image-based Conversation



**1951 USAF
TEST PATTERN
GROUPS 0-3**



Choose Image

Upload

Muffin 4. Whipped Cream Cheese 5. Strawberry Preserves 6. Waffle 7. Flame Roasted Fuji Apples 8. Bagel Blueberry 4oz Unsliced 9. Whole Grain White Bread 10. Sourdough Bread 11. Strawberry Cream Cheese 12. Sunbutter Sunflower Seeds 13. Syrup 14. Strawberry Topping 15. Cinnamon Raisin Bagel 16. Whole Wheat Bread 17. Whipped Margarine 18. Grape Jelly 19. Honey (made locally) 20. Sugar-Free Syrup 21. Semi-Sweet Chocolate Chips These items are listed on a food service website or app interface with interface elements such as filters, a calendar view, and allergen information represented by icons.

这个菜单截图里有什么食物吗？请翻译一下中文，谢谢！

这张图片显示的是一份菜单截屏，菜单来自于“Burge Market Place”，并且着重显示的是“Burge Lunch”（Burge 午餐）的一部分。菜单上列出了各种面包和涂抹品，早餐谷物和华夫饼吧台的选择。以下是食物和一些配料的翻译：
- Plain Bagel - 素百吉饼 - Everything Bagel - 杂粮百吉饼 - English Muffin - 英式松饼 - Whipped Cream Cheese - 打发奶油芝士 - Strawberry Preserves - 草莓果酱 - Waffle - 华夫饼 - Flame Roasted Fuji Apples - 火焰烤富士苹果 - Bagel Blueberry 4oz Unsliced - 4盎司整粒蓝莓百吉饼 - Whole Grain White Bread - 全谷物白面包 - Sourdough Bread - 酸面包 - Strawberry Cream Cheese - 草莓奶油芝士

Thank you so much! Have a nice day!

You're welcome! If you have any more questions in the future, feel free to ask. Have a great day!

What is the weather like in Iowa City today?

I'm sorry, but as an AI, I do not have real-time data capabilities to provide current weather updates. You can check the latest weather for Iowa City by looking at a reliable weather website or app for the most up-to-date information.

I see, how many people in the Iowa City?

According to the United States Census Bureau, the estimated population of Iowa City was around 74,828 as of July 1, 2021. It's always good to check the most recent data for the latest estimates, as population numbers can change over time.

Type your question here...

Send

Webcam is local on your browser, and **no images or keys are saved** by this demo.

Dev'd by [G6 of ECE595 GAIT \(Iowa\)](#)

Figure 6: Image-based Conversation Chatting Only

4 Discussion

Before starting this project, all the group members had limited knowledge or background in web development. As a result, we seek to use ChatGPT 4 to shed some light. To begin with, we told GPT about our plan, that is, building a web application using Python, JavaScript, and HTML. With some prompt engineering, GPT soon gave us a structure for the project: a static folder to store any icon images, CSS files, and JavaScript files; a templates folder to store the HTML files; and a utility folder to store all the APIs that we need to use. Besides, there is an app.py Python file that runs the server. At first, we were not sure how these files interact with each other. However, through chatting with GPT and transitioning from asking GPT to write all the code to writing some code by ourselves, we started to learn the basis of web development. We learned that HTML controls the web page layout. With the help of Bootstrap, we can improve the quality of the widgets and can split content into rows and columns easily. JavaScript controls the logic behind the widgets on the web page and communicates with the app server implemented in Python. The app server calls different APIs to execute the related functions and return the output to the web page through JavaScript. All these understandings cannot be achieved without ChatGPT 4 building an interactive learning atmosphere.

4.1 Ethical Consideration

The speech-based translator requires the recording of the user's voice, which contains sensitive personal information. Users may be concerned about inappropriate usage of their voice information, for example, generating fake voicemails with the help of generative AI tools.

4.2 Future Development of Generative AI tools

Recent generative AI tools rely heavily on prompts. The performance of the generated text or image varies greatly depending on the prompts. It shows that generative AI tools do not think the way humans do. Many users struggle to customize the prompts to make the AI tools work as expected. Sometimes, it even takes more effort to work with AI tools. The hope is to improve the generalization of AI tools to communicate efficiently with humans.

4.3 Technical Issues and Future work

We implemented a conversation assistant that answers random questions or questions about an image. One potential improvement is to actively answer questions regarding the conversation itself. However, this function requires the assistant to keep track of the ongoing conversation. We implemented such an assistant, but it became much more complicated with other GPTs included in our project. We

believe that adding this component would drastically increase the functionality of our translator.

4.3.1 Issues in Image-based Conversation Assistant

Two key issues:

1. Conversation Fails to read the image.
 2. Conversation is forced to read the image.
- Conversation Fails to read the image.

As shown in Fig.7, the AI assistant fails to read the image when the user asks a question about the image. This issue is because at first, we use the initial description when the image is submitted plus the user message as a prompt in the ChatGPT API function calling, using the method: `response = client.chat.completions.create(...)`. But actually, the ChatGPT API could not relate the description with the image, and what's more, the description is unable to fully cover all the details of the image. Thus, in other words, the user message question should reach out to the image every time to guarantee a good answer. So, we change the using method: `payload = (...)`, to call both image plus user message parameters and then to fix it.

- Conversation is forced to read the image.

As shown in Fig.8, the AI assistant is forced to read the image when the user asks a question irrelevant to the image, such as "Thank you!". This issue is because the issue fixed above would make the user message input strongly connect to the image submitted. Thus, we add a function: `def isImageRelated()` to detect whether the user message input is individual or related to the image automatically. If it is not related, the ChatGPT API function will call back to `response = client.chat.completions.create(...)` to achieve just chatting without an image.

Translation Output for User 1

Translation Output for User 2

WebcamGPT-Vision

The image shows a **1951 USAF (United States Air Force) Test Pattern** which is used for resolving power measurements on imaging systems. This test chart includes sets of **parallel lines** at various widths and spacings, organized in **groups** and **elements** labelled with numbers along the left side and top. In the middle, there is a **central square** with an intricate **checkerboard pattern**. At the bottom of the image, the text reads "**1951 USAF TEST PATTERN GROUPS 0-3**," indicating the range of groups depicted in this particular section of the chart.

Objects:

- 1951 USAF Test Pattern
- Parallel lines
- Groups and elements labels
- Central square with a checkerboard pattern
- Text description

Capture Switch Camera

Choose File No file chosen

Upload

Hi

Hello! How can I assist you today?

How many number can you find in the image?

I'm sorry, but it seems there has been a misunderstanding. As an AI language model, I'm not able to directly view or analyze images. However, if you describe an image to me or provide information about it, I can use that description to help answer your questions about the image. If you want to know more about the content of a specific 1951 USAF Test Pattern, you can mention the numbers visible in the groups and elements in your description, and I can provide

Figure 7: Conversation Fails to read image

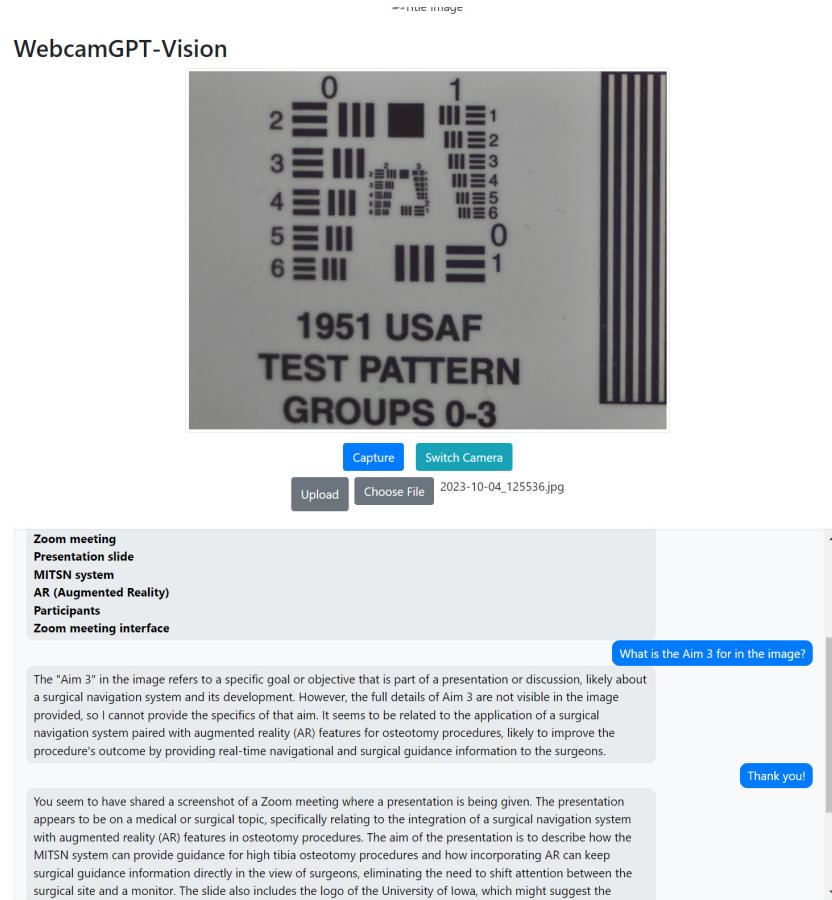


Figure 8: Conversation is forced to read image

4.4 Contribution

Yaohua Wang

- Implemented the speech-to-speech translation.

Yijie Zhang

- Implemented Image-based Conversation Assistant.

Yan Chen

- Conversation assistant to track chat history and answer text-based questions (Failed to incorporate into the pipeline)
- Report writing (Introduction, discussion and workflow) and video editing