VQAsk: a multimodal Android application to help blind users visualize pictures

Clizia Giorgia Manganaro manganaro.2017897@studenti.uniroma1.it
Chiara Giacanelli giacanelli.1801145@studenti.uniroma1.it
Alessio Palma palma.1837493@studenti.uniroma1.it
Davide Santoro santoro.1843664@studenti.uniroma1.it

Computer Science Department, Sapienza University of Rome
Multimodal Interaction Project

What is Visual Question Answering?

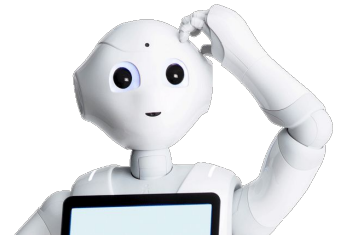
Visual Question Answering (VQA) is a **computer vision task where a system is given a text-based question about an image, and it must infer the answer.**

Why it's important?

- to help **blind users** to communicate through pictures;

- to **attract customers** of online shopping sites by giving "semantically" satisfying results for their search queries;

- **Visual Dialogue**, which aims to give natural language instructions to robots.

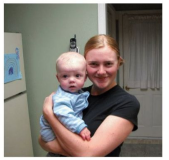


Who is wearing glasses?

man woman

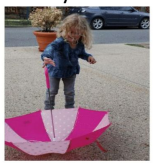
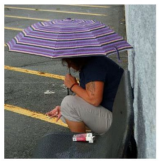

Where is the child sitting?

fridge arms

Is the umbrella upside down?

yes no

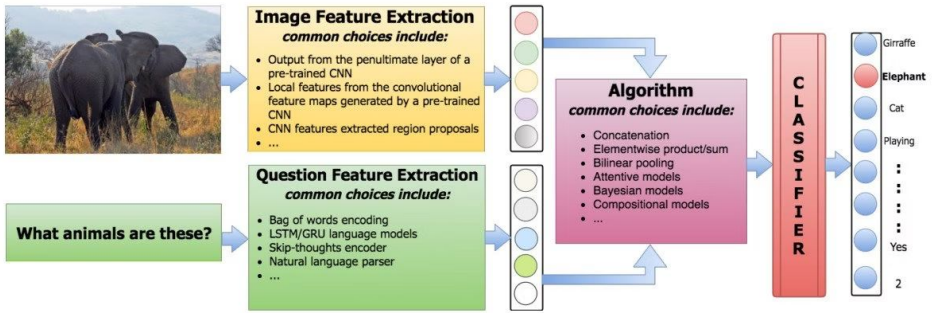



How many children are in the bed?

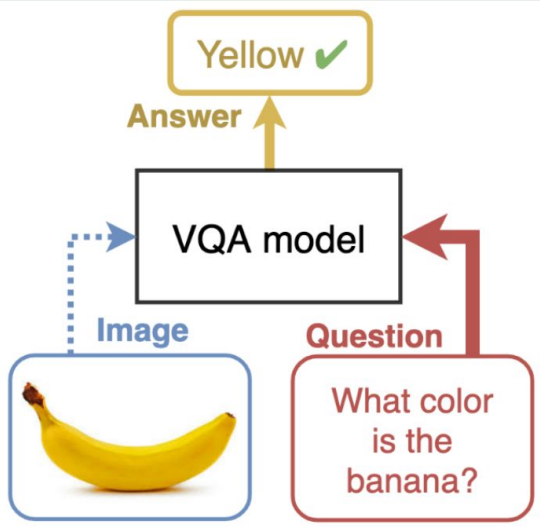
2 1

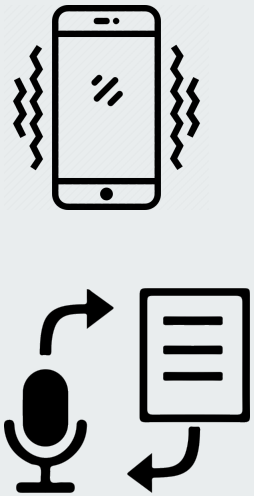

VQA is challenging because...

- It is a multimodal task by definition
- The questions are not predetermined
- The supporting visual information is very high dimensional
- VQA necessitates solving many computer vision sub tasks (such as object detection, activity recognition and scene classification)



Our objectives





1) **Integrate a VQA model** with multimodal interaction

2) Implement a system to help visually impaired (and possibly also blind) people visualize pictures through a mobile application that uses **speech interaction & haptic feedbacks** (as well as the normal touch-screen to type and observe).

3) **Evaluate** the results.

Tools and Technologies Used

This project is built using **Flutter**, an open-source framework developed by Google.

It should be compatible with **SDK versions** greater than or equal to **3.0.6** but less than **4.0.0.**

For Android development, this project targets **Android API version 34**, which allows it to take advantage of the latest Android features and optimizations.

User-Functional Requirements		
ID	DESCRIPTION	MoSCoW
0	The user must be able to load pictures either from the gallery or the smartphone's camera	Must have
1	The user must be able to activate the system with his voice	Must have
2	The user must be able to ask questions about his pictures using the touch-screen keyboard or by voice	Must have
3	The user must be able to read and to listen the answer given by the system	Must have
4	The user must be able to receive instructions about the usage of the system	Should have
5	The user must be able to erase his questions and change it	Must have
6	The user must be able to listen to the inserted question and the given answer whenever he needs	Should have
7	The user must be able to edit his pictures to highlight important portions of them	Should have
8	The user must be able to edit his pictures also by voice	Should have

Requirement Analysis

Functional Requirements

System-Functional Requirements		
ID	DESCRIPTION	MoSCoW
0	The system must provide touch-screen interaction, voice interaction and haptic feedbacks	Must have
1	The system must show clearly all the elements that allow the user to listen and type quesions and answers	Must have
2	The system must allow the user to select pictures from the gallery and to shoot them from the camera	Must have
3	The sytem must show if there are errors in the typing of the question	Should have
4	The system must include an accurate speech recognition module to convert spoken language into text and should support various accents, dialects, and languages for robust speech processing	Must have
5	The system should be able to associate spoken or written questions with the relevant features extracted from the images	Must have
6	The system should allow the user to edit the images	Should have

Requirement Analysis

Functional Requirements

Non-Functional Requirements		
ID	DESCRIPTION	MoSCoW
0	The application has to work in Android v.11 systems (or higher)	Must have
1	The images have to be processed securely	Must have
2	The application must be able to process .jpg and .PNG formats	Must have
3	The application must be able to ask the permission to the user to activate the microphone and the camera	Should have
4	The application must quit listening if the user stops pronouncing words	Should have
5	The application must provide an answer in a reasonable time	Must have

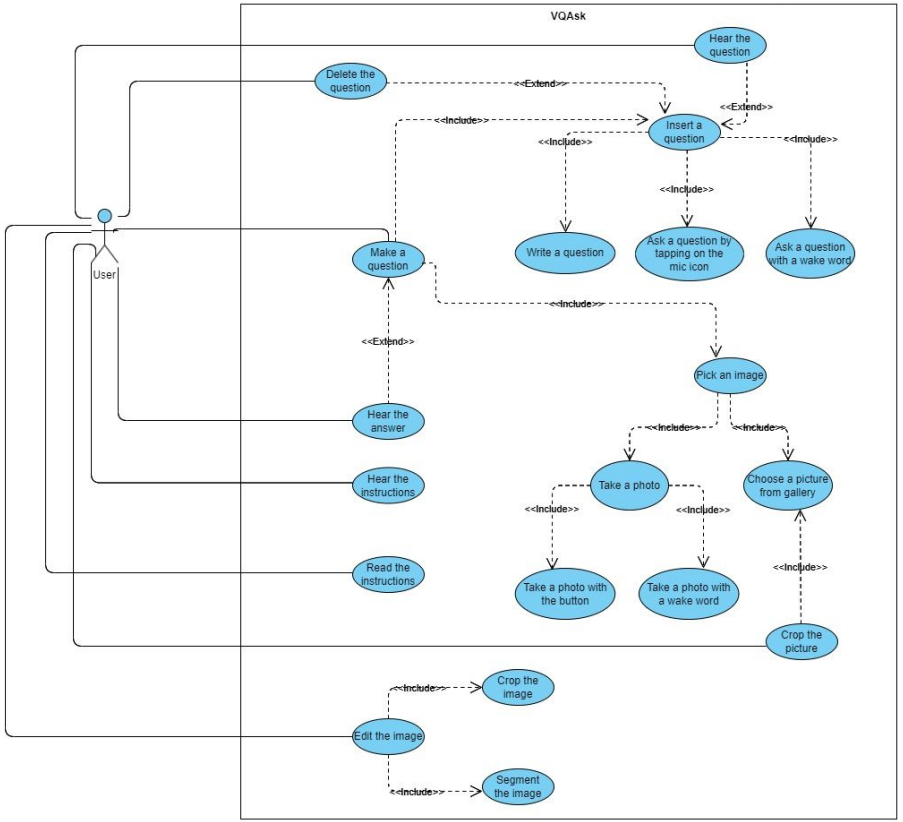
Requirement Analysis

Non-Functional Requirements

UML diagrams

In order to well document the system functionalities we made some UML diagrams:

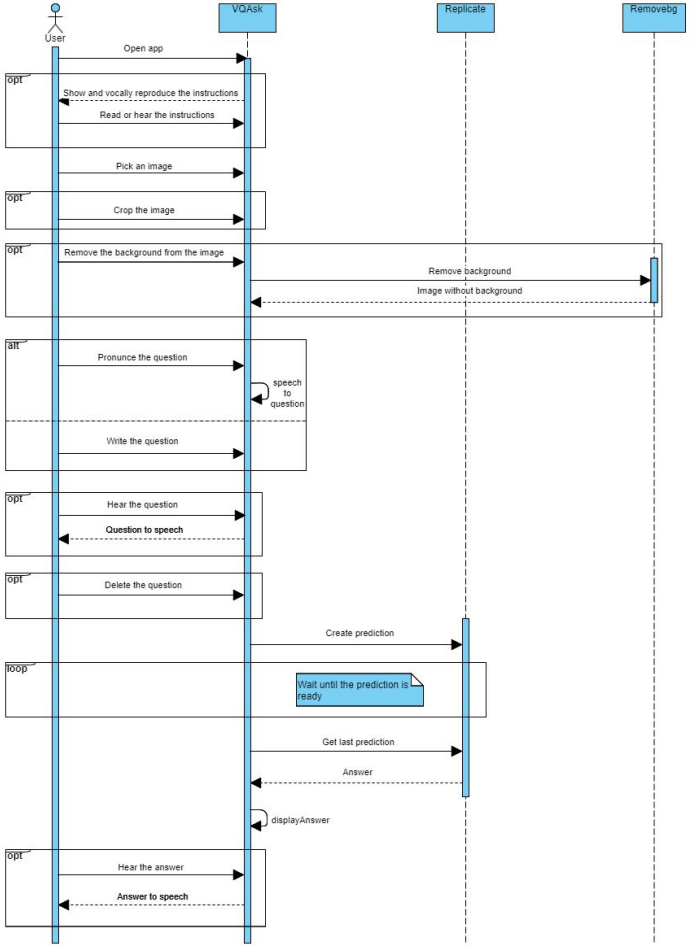
- **Use-case diagram** to describe the high level functions of the system
- **Sequence Diagram** for the inner workings of the **“standard” interaction mode**
- **Sequence Diagram** for the inner workings of the **“alternative” interaction mode**



UML diagrams (cont'd)

Sequence diagram that represents the **"standard" interaction mode** with the system

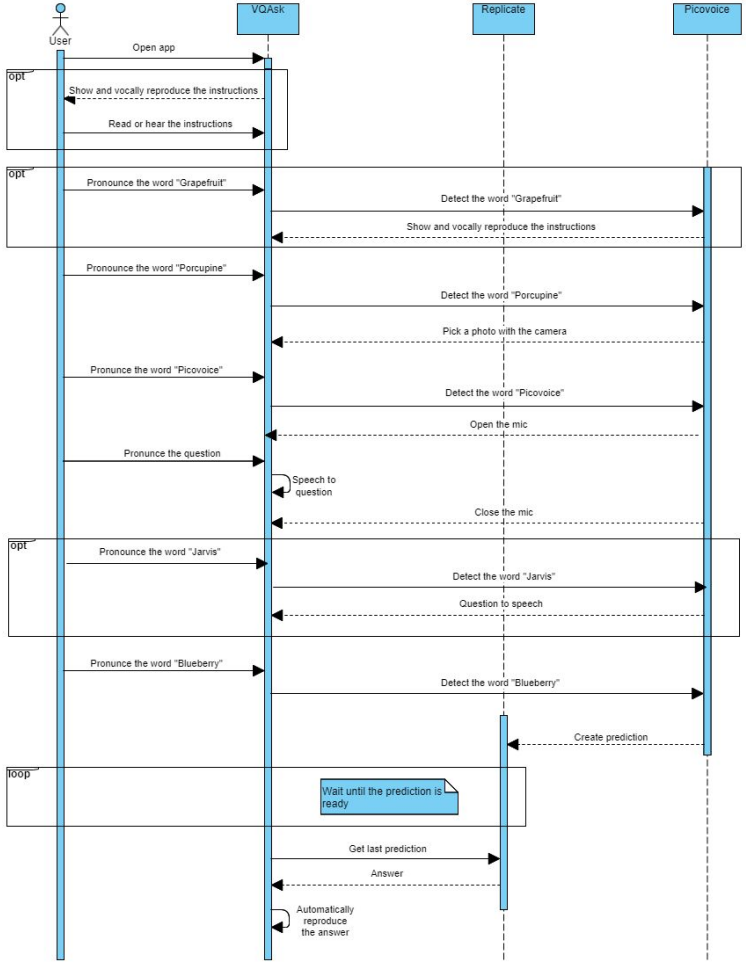
- Mainly based on visual interaction
- Includes the following actors:
  - User
  - VQAsk
  - Replicate
  - Removebg



UML diagrams (cont'd)

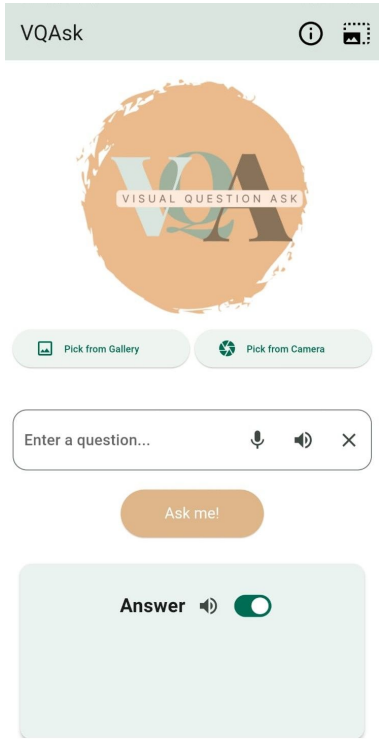
Sequence diagram that represents the **"alternative" interaction mode** with the system

- Based exclusively on vocal interaction
- Includes the following actors:
  - User
  - VQAsk
  - Replicate
  - Picovoice

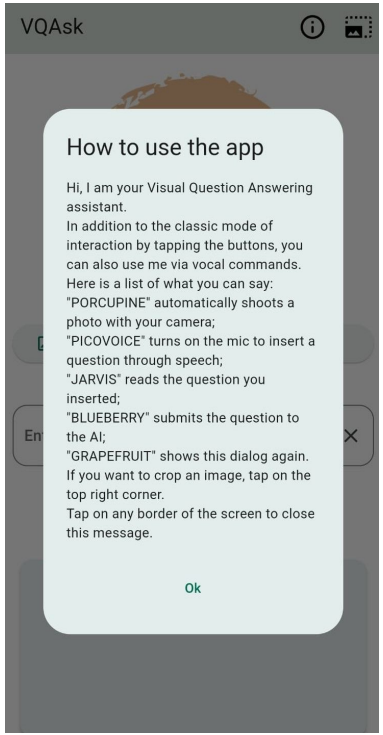


Building the Application (1)

Homepage



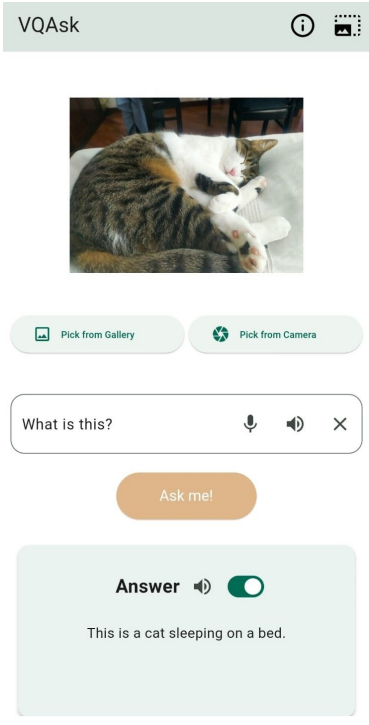
Info Dialog



Semaphoric Words

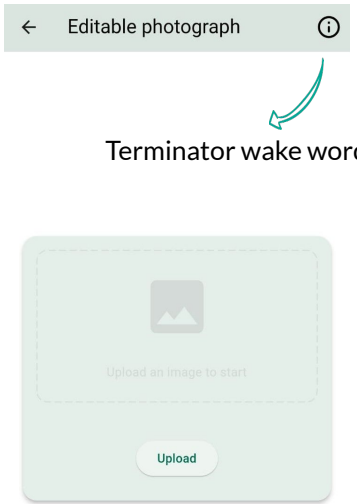
- **PORCUPINE**: the application automatically captures an image from the external camera of the mobile phone, and loads it in order to be used by the app;
- **PICOVOICE**: the following sentences will be recorded in order to be used as the question related to the image. The user's speech can be continuous and the speaking style spontaneous;
- **JARVIS**: the app will output the inserted question by voice;
- **BLUEBERRY**: the question will be submitted in order to be answered;
- **GRAPEFRUIT**: the Info Section will be opened and read again by the Vocal Assistant:

Asking questions: step



Building the Application (2)

Editable photograph page



Functionalities

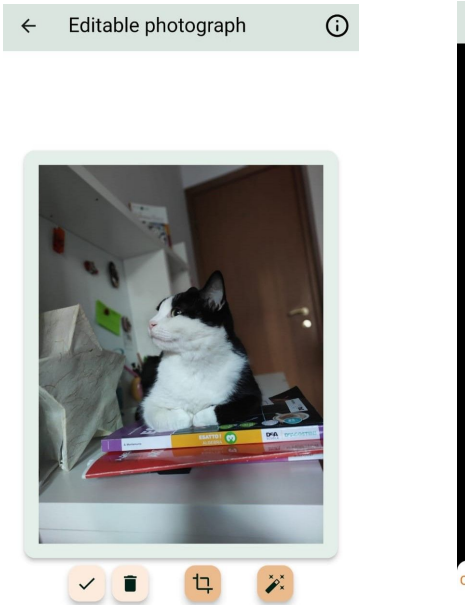
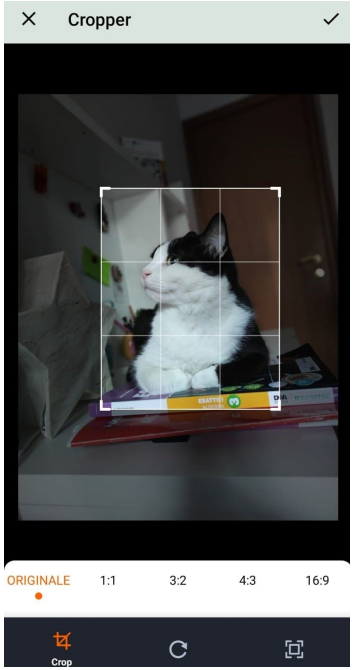


Image Cropping

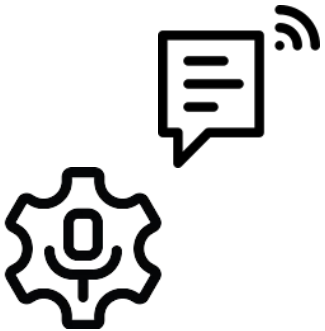


AI segmentation

Designing Multimodal Interaction

Voice Interaction:

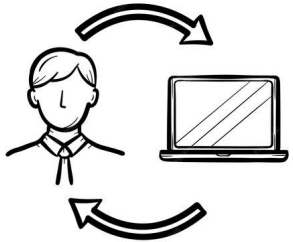
1. text-to-speech
2. speech-to-text



Haptic Feedbacks
provide device vibrations
as feedbacks

Visual Interaction:
thanks to the icons

Grounding



- Establish a **common understanding** of the system's state between user and system, of crucial importance in the case of visually impaired users
- Achieved through visual, auditory and haptic feedback
- Standard feedbacks that can be well understood by everyone with **minimal experience**
- Some examples:
    - Short vibration ⟹ CONFIRMATION, long vibration ⟹ ERROR
    - *"I'm thinking"* when loading the answer
    - Shutter release sound when a photo is taken

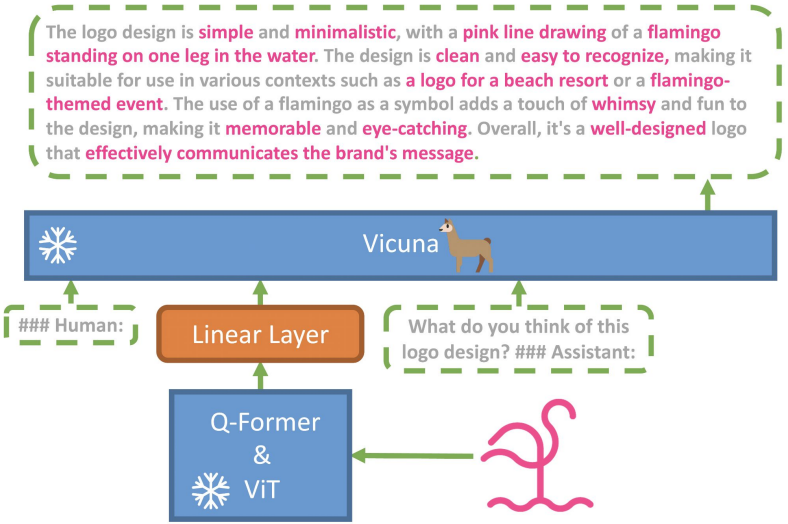
From LLMs to VLMs

Large Language Models (**LLMs**) represents a paradigm shift in NLP. They are, at their core, **Transformer architectures** with billions of parameters and train data, meticulously designed to process and generate human language with unparalleled competence.

Vision-Language Models (**VLMs**): multimodal powerhouses, fusing text with other data modalities such as images or audio. Comprehend, generate, and manipulate both textual and visual information seamlessly, introducing multimodality inside the neural network!

MiniGPT-4

- **GPT-4:** closed source, APIs are fee-based, not a feasible choice
- MiniGPT-4: open source, uses Vicuna (built upon LLaMA by Meta) as the language decoder, ViT + Q-Former as the visual encoder. Only trains a single projection layer to align the encoded visual features with the language model, freezing all the other components

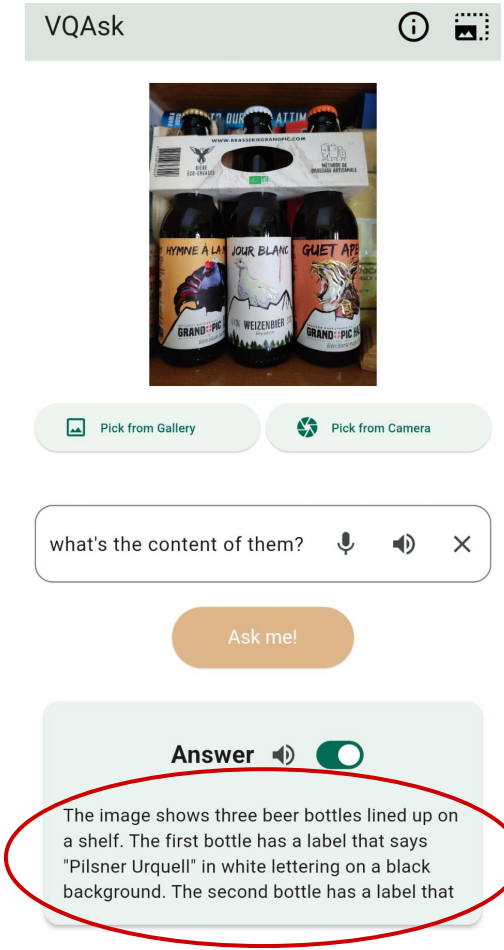


MiniGPT-4 (2)

Like all AI models today, it still faces several limitations:

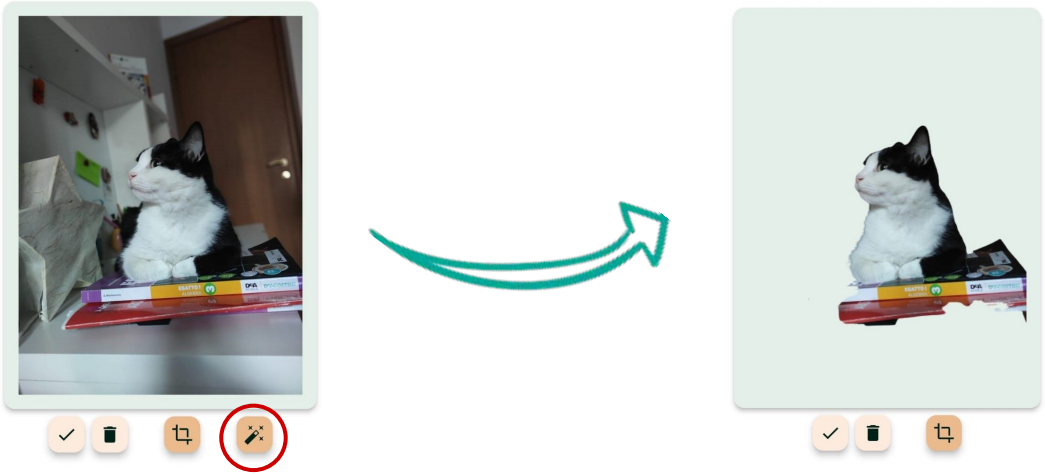
- **Language hallucination**: as MiniGPT-4 is built upon LLMs, it inherits LLM's limitations like unreliable reasoning ability and hallucinating nonexistent knowledge
- **Inadequate perception capacities:** MiniGPT-4's visual perception remains limited. It may struggle to recognize detailed textual information from images, and differentiate spatial localization

Do not take it as a source of absolute truth!



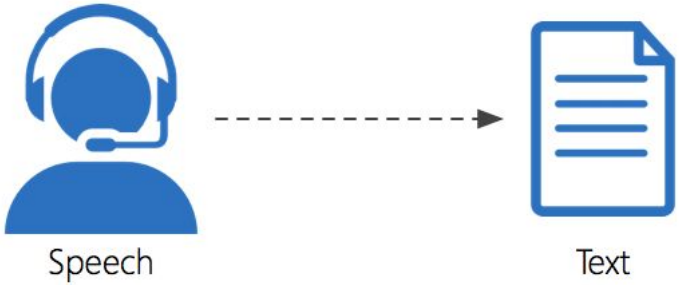
Remove.bg

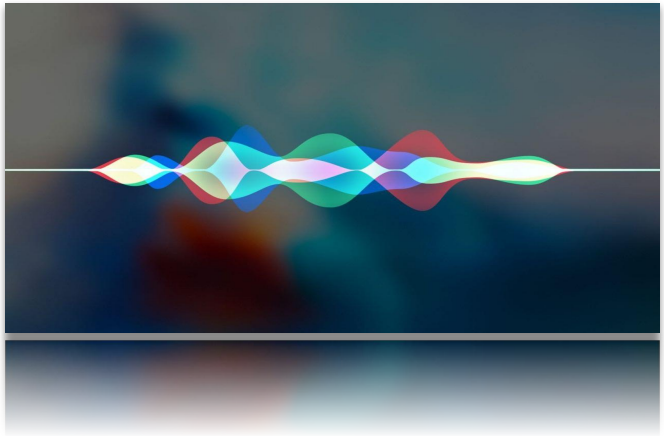
In order to segment the main object in the images and enhance the VQA performance we have used Remove.bg APIs. It is a web-based segmentation service, designed to eliminate the background from any photo. It employs closed-source AI technologies.



Speech-to-Text

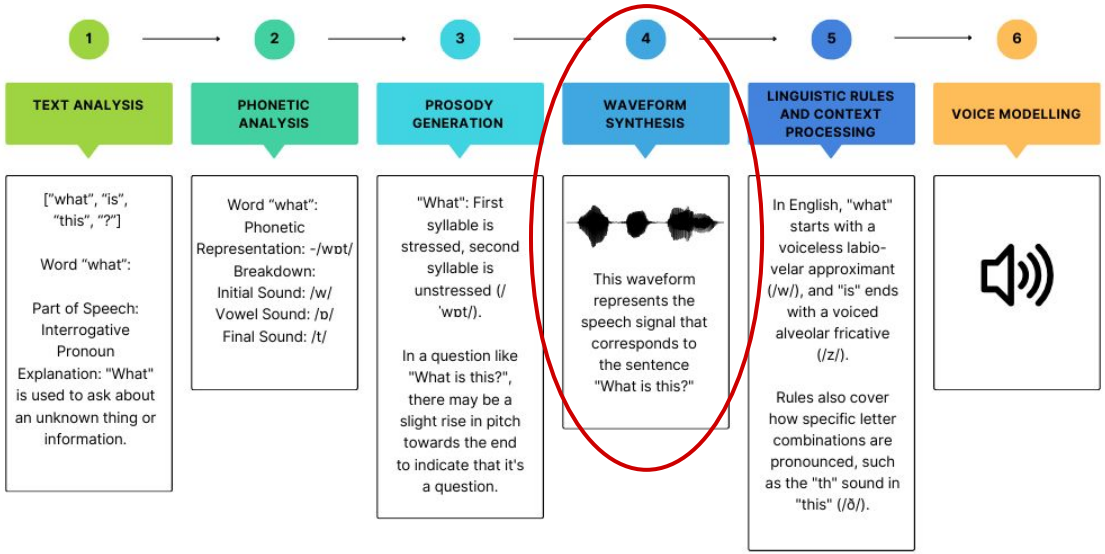
- speech_to_text is a Flutter library that exposes device specific speech-to-text recognition capabilities
- By converting spoken words into text users can, for example, enter the question by dictating it without using the keyboard
- This feature enhances accessibility, allowing also individuals with physical disabilities to use the app
- It leverages the native voice recognition engines provided by the Android and iOS operating systems
- 3 main approaches: acoustic-phonetic, pattern recognition, AI based





Text-to-Speech

Flutter_tts is a popular Flutter library to integrate TTS functionalities into Flutter applications. It is again a wrapper around the native text-to-speech engines provided by the device's operating system. A common workflow is:



3 main approaches:

- concatenative synthesis
- formant synthesis
- articulatory synthesis

Testing & Evaluation

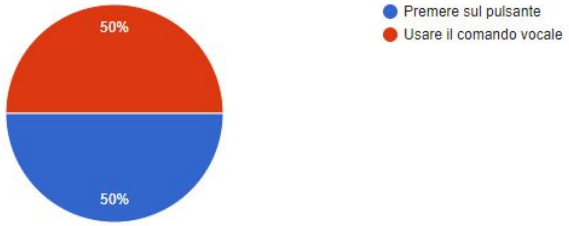
- Sample of **10 users**
- We asked them to complete a series of **pre-established task**
- Tests were performed by using the **Think-Aloud** methodology
- At the end of the test, we asked users to fill in a **short questionnaire**

Testing & Evaluation (2)

Some questionnaire results about the preferences of the users

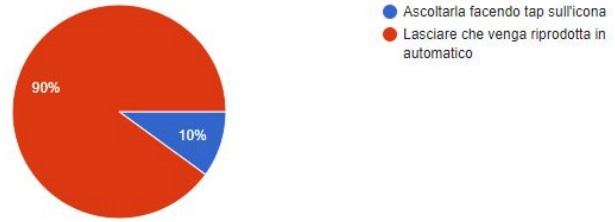
Per inviare la domanda al sistema hai preferito:

10 risposte



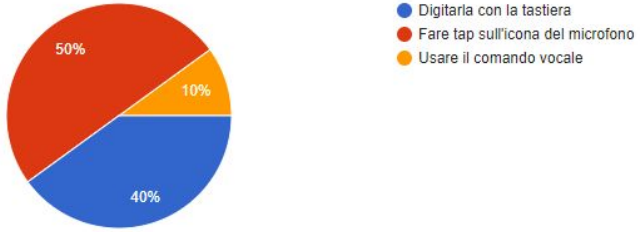
Per ascoltare la risposta hai preferito:

10 risposte



Per inserire la domanda hai preferito:

10 risposte

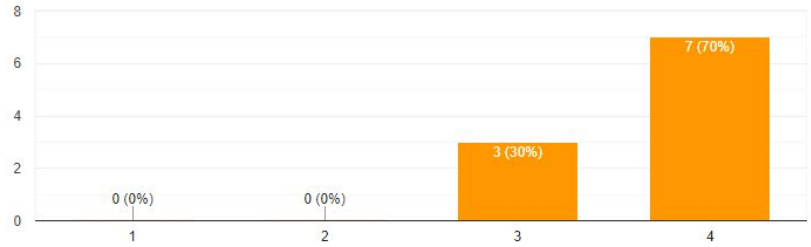


Testing & Evaluation (3)

Some questionnaire results about users experience and use of the app

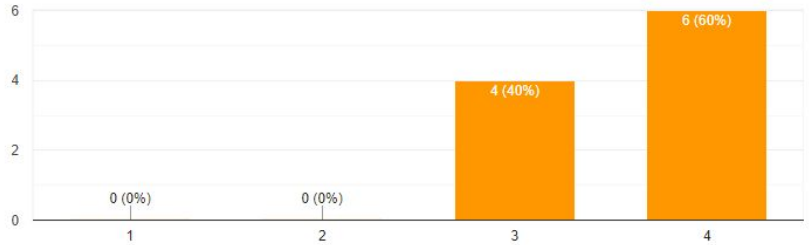
Quanto pensi che questa applicazione possa essere utile in contesti reali?

10 risposte



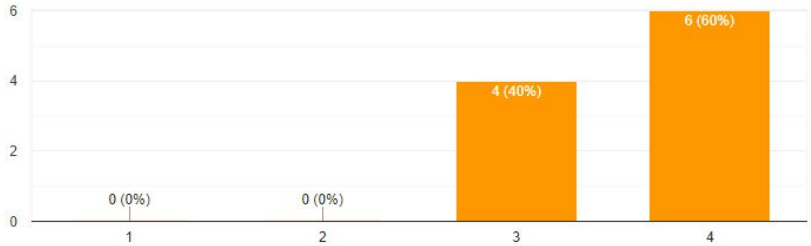
Come valuti l'esperienza generale di utilizzo dell'app:

10 risposte



Pensi che saresti in grado di usare l'applicazione se fossi ipovedente (quindi usando solo l'interazione vocale)?

10 risposte



Conclusions

Our Android application's successful integration of voice interaction, vision utilization, and haptic feedback, coupled with NLP and Computer Vision techniques, positions it as a **valuable solution for improving accessibility** and enhancing the overall user experience for individuals with visual impairments.

Future possible improvements:

1. introducing **additional features** that extend the application's utility across various domains;

2. incorporating **acoustic interactions**, such as beeps or other auditory cues;

3. exploring the adaptation of this application onto **specialized hardware** designed to assist individuals with visual difficulties in their everyday lives, creating a dedicated device → test it on visually impaired users!

Thanks for your attention!

And now let's move to the demo...