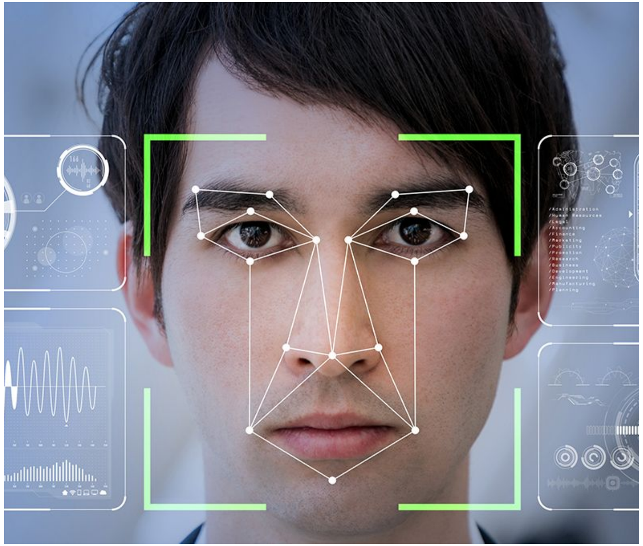
Multimodal YouTube

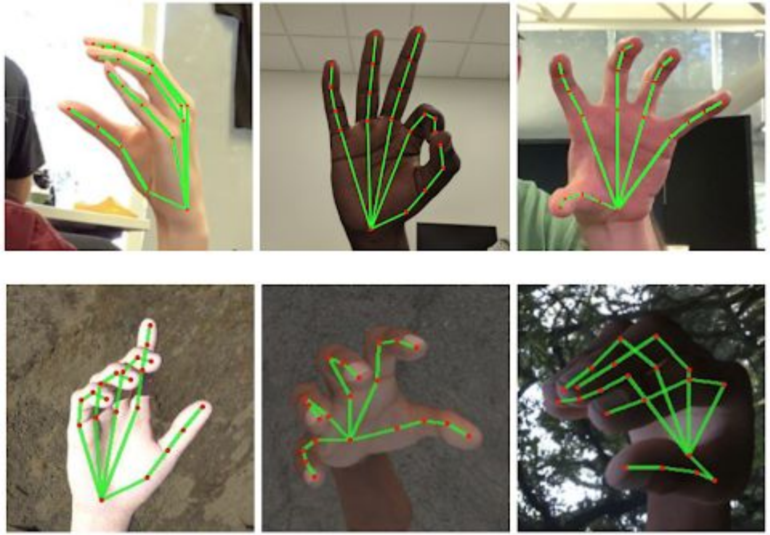
Projects of Biometric Systems and Multimodal Interaction

by Irene Cannistraci & Giovanni Ficarra

Introduction

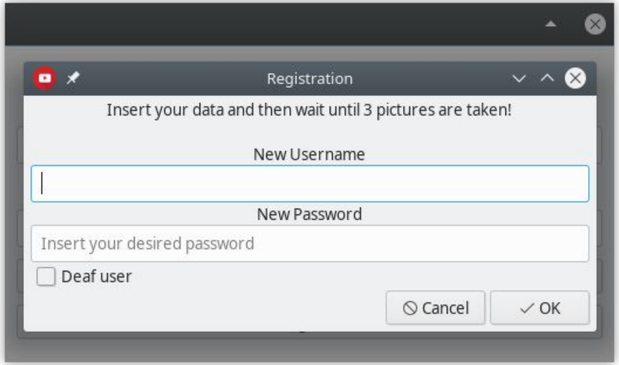
Multimodal YouTube is an application designed to be accessible by both deaf users and not, that combines **biometric** and **multimodal** methodologies.





Flow and Architecture

1. If not enrolled yet, the user can **register**;
2. If already registered, the user can **login** using *face recognition* or typing username and password;



Registration

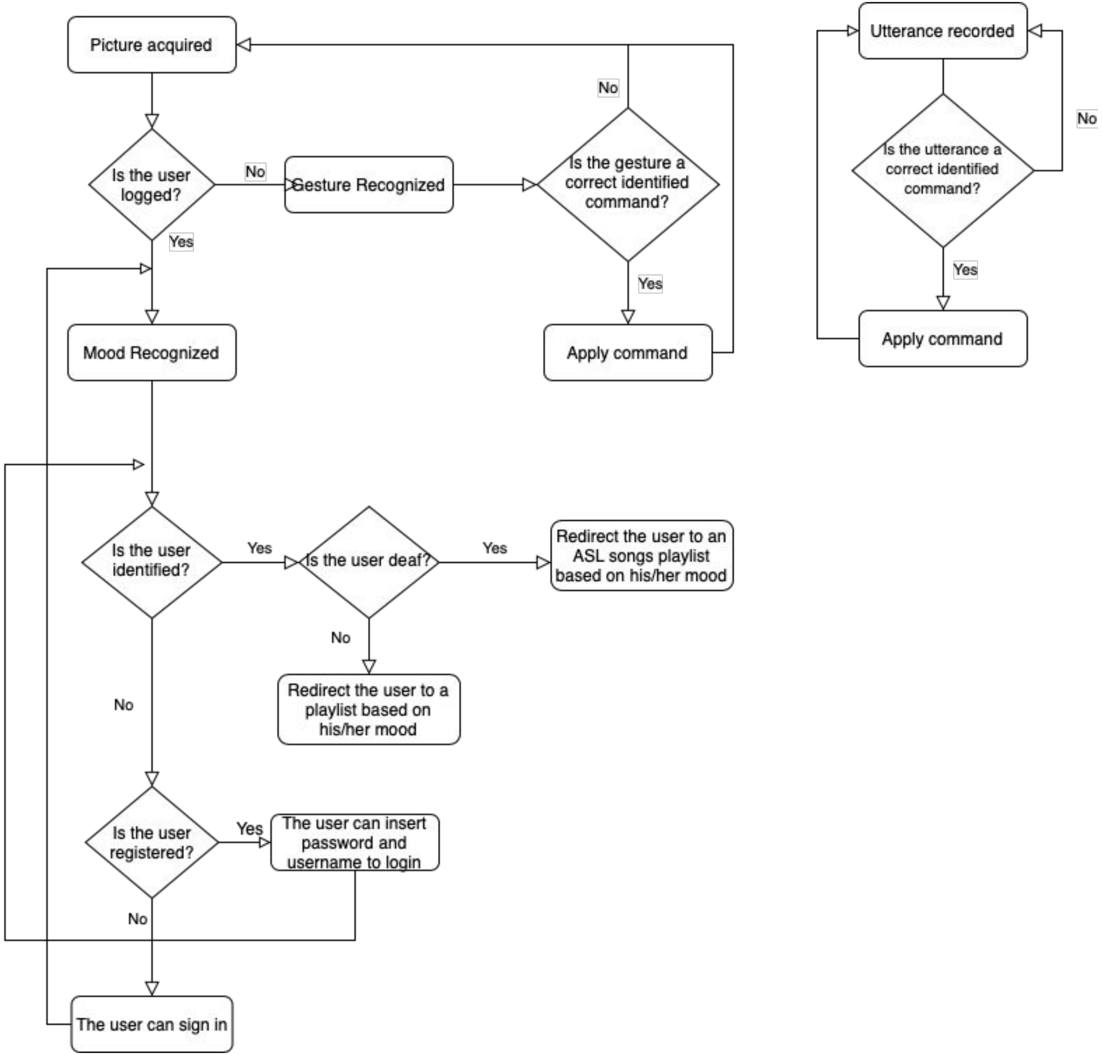
Insert your data and then wait until 3 pictures are taken!

New Username

New Password

Insert your desired password

☐ Deaf user



Picture acquired
Is the user logged?
No
Gesture Recognized
Is the gesture a correct identified command?
No
Yes
Apply command
Yes
Mood Recognized
Is the user identified?
Yes
Is the user deaf?
Yes
Redirect the user to an ASL songs playlist based on his/her mood
No
Redirect the user to a playlist based on his/her mood
No
Is the user registered?
Yes
The user can insert password and username to login
No
The user can sign in
Utterance recorded
Is the utterance a correct identified command?
No
Yes
Apply command

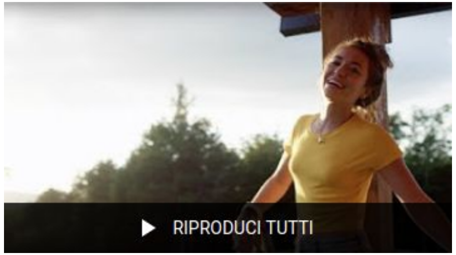
Flow and Architecture

1. If not enrolled yet, the user can **register**;
2. If already registered, the user can **login** using *face recognition* or typing username and password;
3. The system use the *emotion recognition* and the deaf information to **redirect** the user to the correct playlist;

ASL Happiness

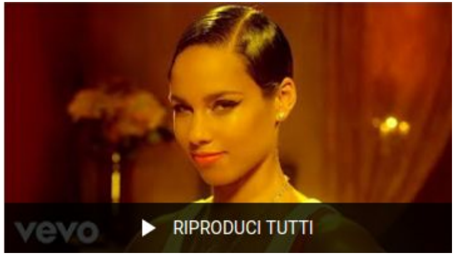
Happiness

ASL Sadness



ASL Sadness

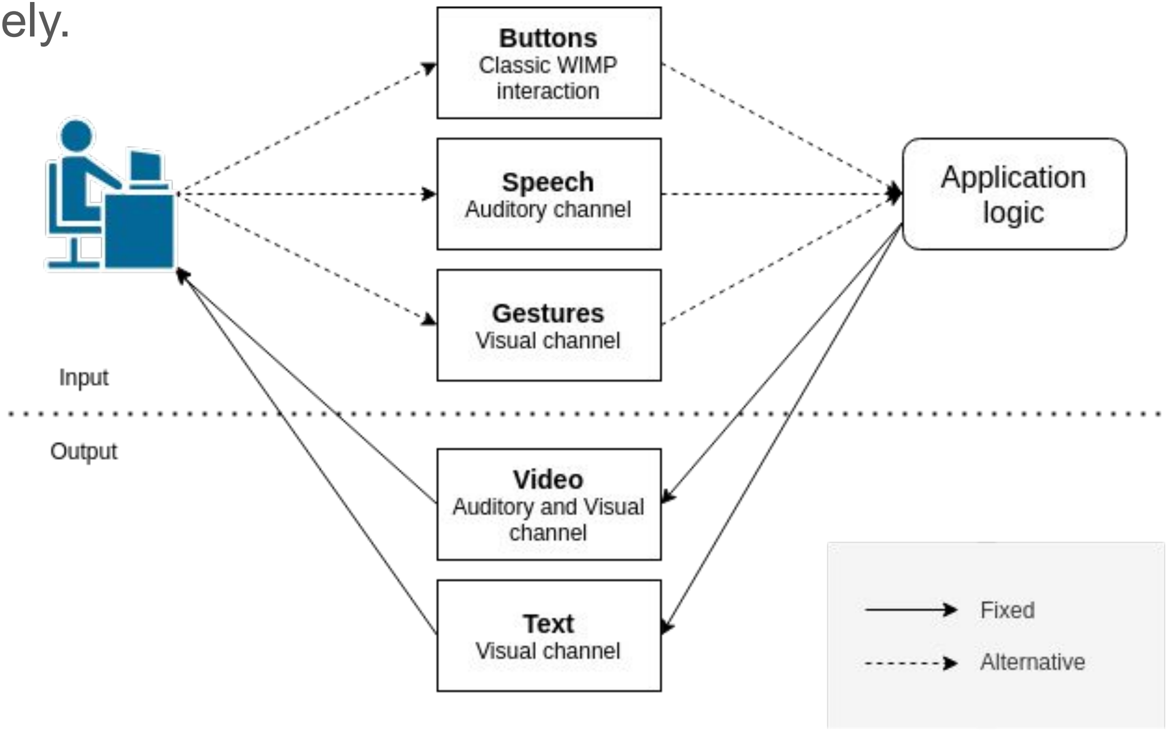
ASL Neutral



ASL Neutral

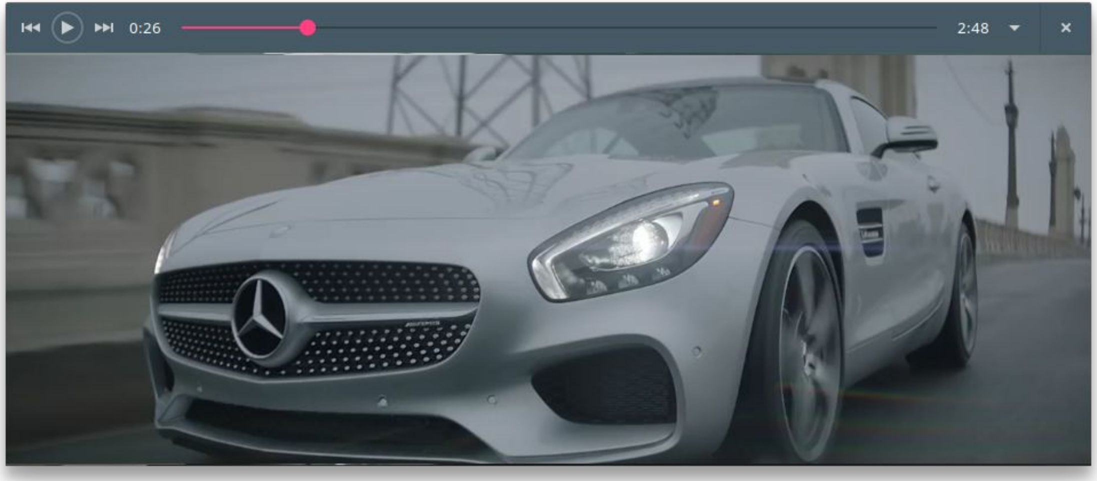
Flow and Architecture

4. The user can **interact** with the player using buttons, *gestures* or *speech*, alternatively.



API & Services

- **Vishnunarayan K I’s YoutubePlayer**:
    - nice and simple interface,
    - support for YouTube playlists,
    - based on GTK toolkit.

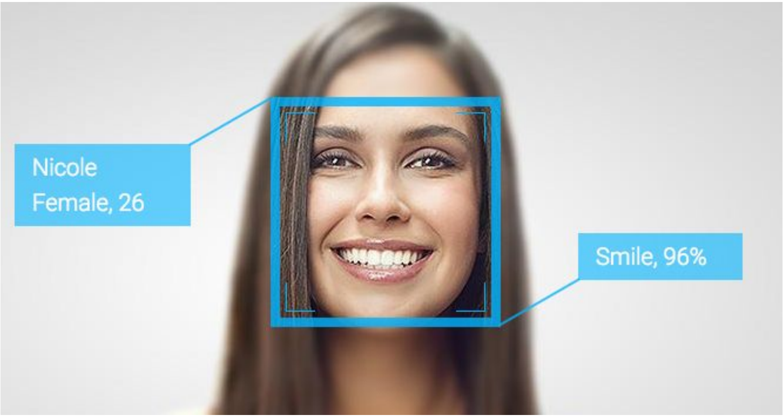


API & Services

- **Vishnunarayan K I’s YoutubePlayer**;
- **FACE++**:
  - *Gesture Recognition API*:
    - send the picture with the gesture to the API,
    - receive the detected and classified gesture(s).

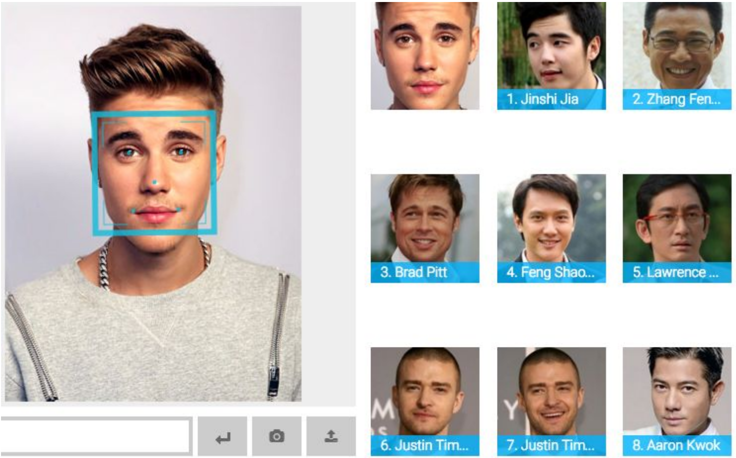
API & Services

- **Vishnunarayan K I's YoutubePlayer**;
- **FACE++**:
  - *Gesture Recognition API*,
  - *Face Detection API*:
    - send the picture with the face to the API,
    - receive the *face token* and the recognized *emotion,*
    - during login, save the face in a cloud gallery (*faceset*) and the token in the DataBase.



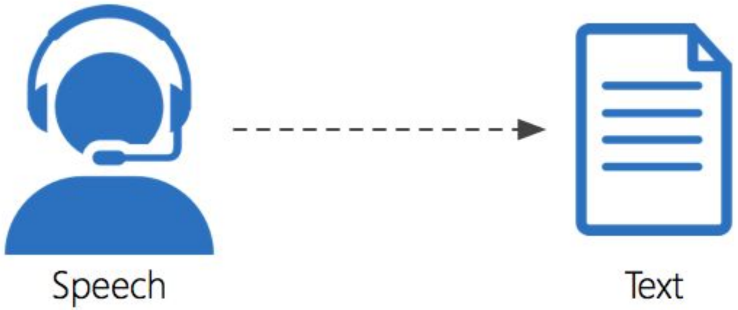
API & Services

- **Vishnunarayan K I’s YoutubePlayer**;
- **FACE++**:
  - *Gesture Recognition API*,
  - *Face Detection API*,
  - *Face Searching API*:
    - obtain a new face token with Face Detection API,
    - send the token to the API, which will compare the relative face with those in the specified faceset,
    - receive the face of most similar face, the relative confidence and three thresholds.



API & Services

- **Vishnunarayan K I’s YoutubePlayer**;
- **FACE++**;
- **Microsoft Azure Cognitive Services:**
  - *Speech to Text API* quickly transcribes audible speech into readable text,
  - create a new speech recognizer session,
  - the system will start continuously listening and recognize the real-time audio in order to translate it into text.



Evaluation - The dataset

We generated three custom datasets where users have various ages, english proficiency and devices:

- *Identification_and_Emotion*: 31 users with ≥ 3 pictures each for the three possible emotions (neutral, sad, happy) - 103 pictures;
- *Gesture_dataset*: 30 users with ≥ 6 pictures each for the six possible gestures (hand open, thumb up, thumb down, index finger up, victory, fist) - 187 pics;
- *Speech_dataset*: 31 users with 1 audio each, in which they pronounce all the possible voice commands (play, pause/stop, up, down, skip/next, previous, mute, unmute) - 31 audios.

Evaluation - Open Set Identification

We split the Identification_and_Emotion dataset into 25 *genuine users* and 6 *impostors*.

We performed a first test with the i**ntermediate threshold 71.8%**, and we got:

- *DIR* = correct_matches/n_genuine = **1** (the system was able to recognize all the genuine users);
- *FRR* = 1-DIR = **0** (the system had never reject a genuine user);
- *FAR* = false_alarms/n_impostors = **0.44** (the system had accepted users that were impostors).

Evaluation - Open Set Identification

We performed a second test with the **highest threshold 76.5%**, and we got:

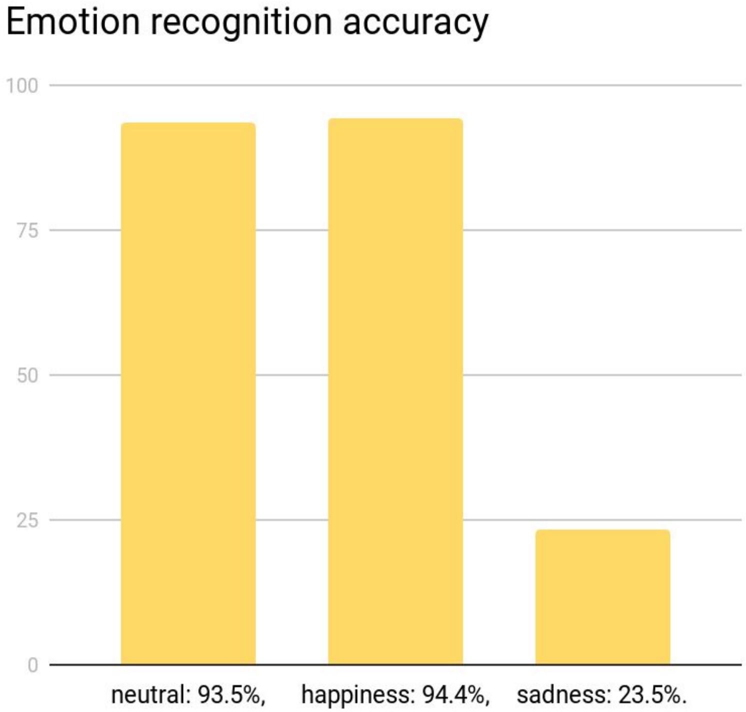
- *DIR* = 5/33 = **0.15** (the system was not able to recognize most of the genuine users);
- *FRR* = 1 - 0,15 = **0.85** (the system reject almost all the genuine users);
- *FAR* = 18/20 = **0.9** (the system had accepted just a few users that were impostors).

Since in this way the system rejects most of the genuine users, we decided to restore the intermediate threshold.

Evaluation - Emotion Recognition

We tested the system on all the 109 available pictures and we got

Accuracy = correct_emotion/n_faces = **69%.**



Evaluation - Emotion Recognition

We tested the system on all the 109 available pictures and we got

Accuracy = correct_emotion/n_faces = **69%.**

Looking at the single scores we notice that:

- **Happiness** was the most recognized expression: **94.4%**,
- **Sadness** was the least recognized emotion, often misclassified as *neutral*: **23.5%**.

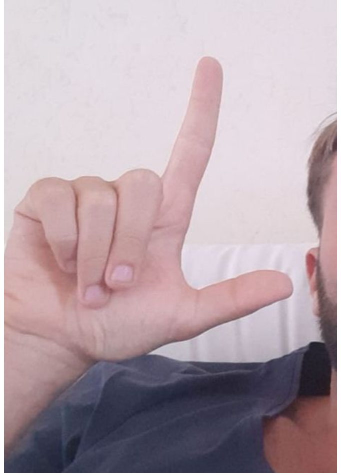
Evaluation - Gesture Recognition

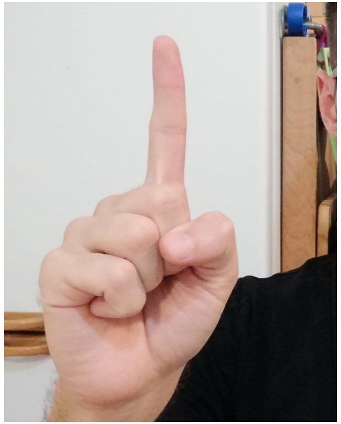
We performed tests on all the 187 available pictures and we got:

- *Accuracy* = correct_gestures/n_photo_gestures = 110/187 = **58.82%**

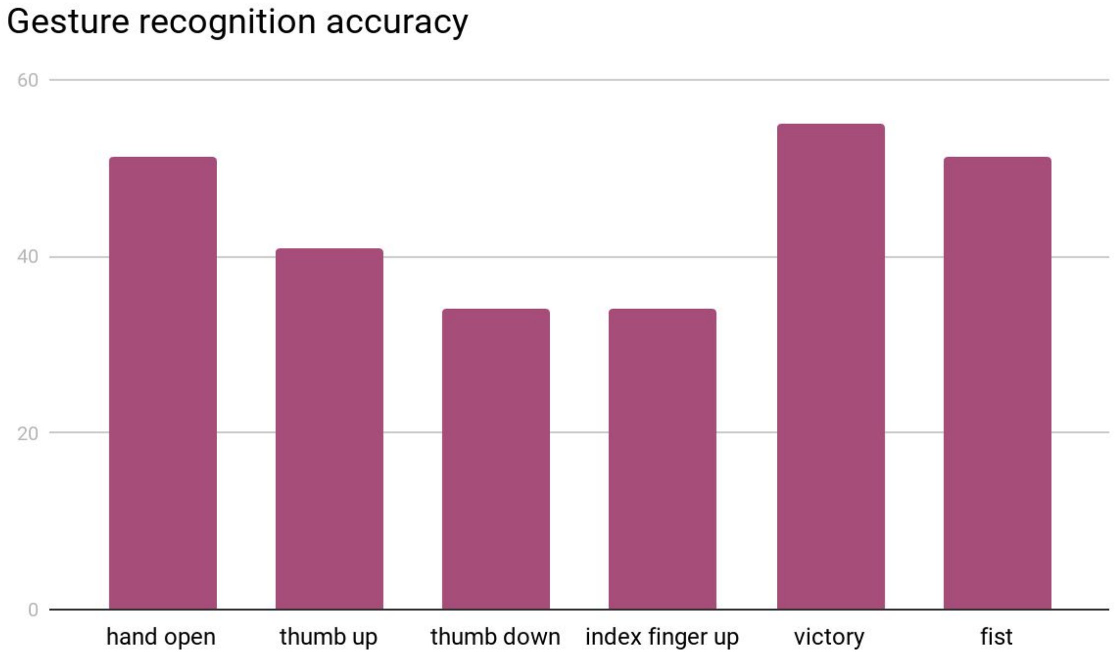
Looking at the results we notice that the total number of hands identified by the system was **252**, **65** more than they should be. We also noticed that:

- **index finger up** was the most hard to recognize: a possible explanation could be that a lot of users made a *wrong gesture* since they created an “L” with their hand, instead of raising only the index finger: **34.04%**;
- **victory** is most one recognized, but the accuracy is low too: **55%.**





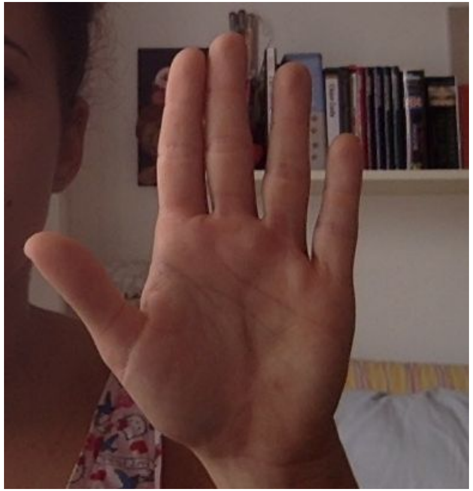
Evaluation - Gesture Recognition

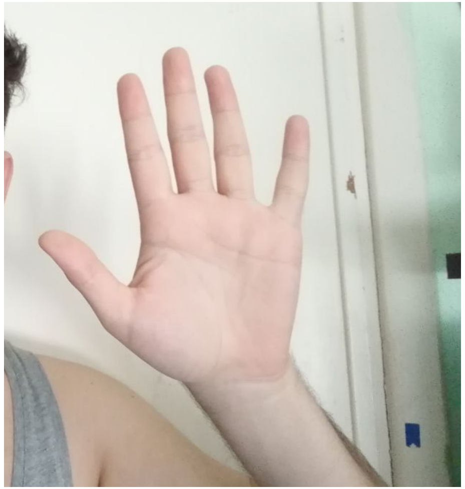


Evaluation - Gesture Recognition

We noticed that the results are strictly user-dependent:

- cluttered or contrasting background,





Evaluation - Gesture Recognition

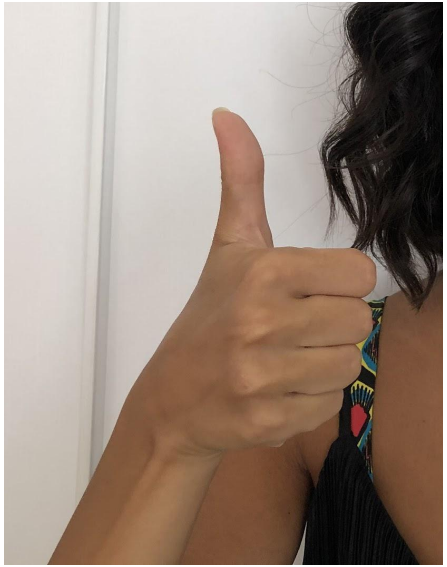
We noticed that the results are strictly user-dependent:

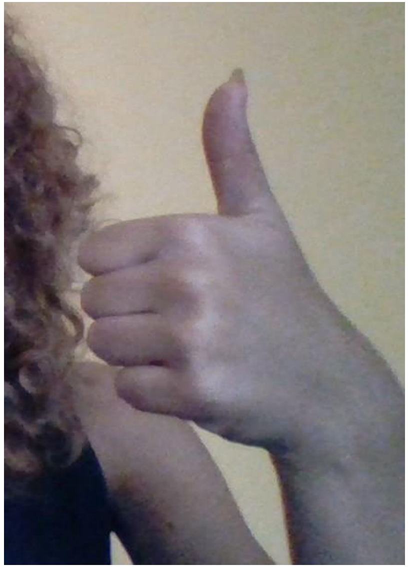
- cluttered or contrasting background,
- visible user or only hand

Evaluation - Gesture Recognition

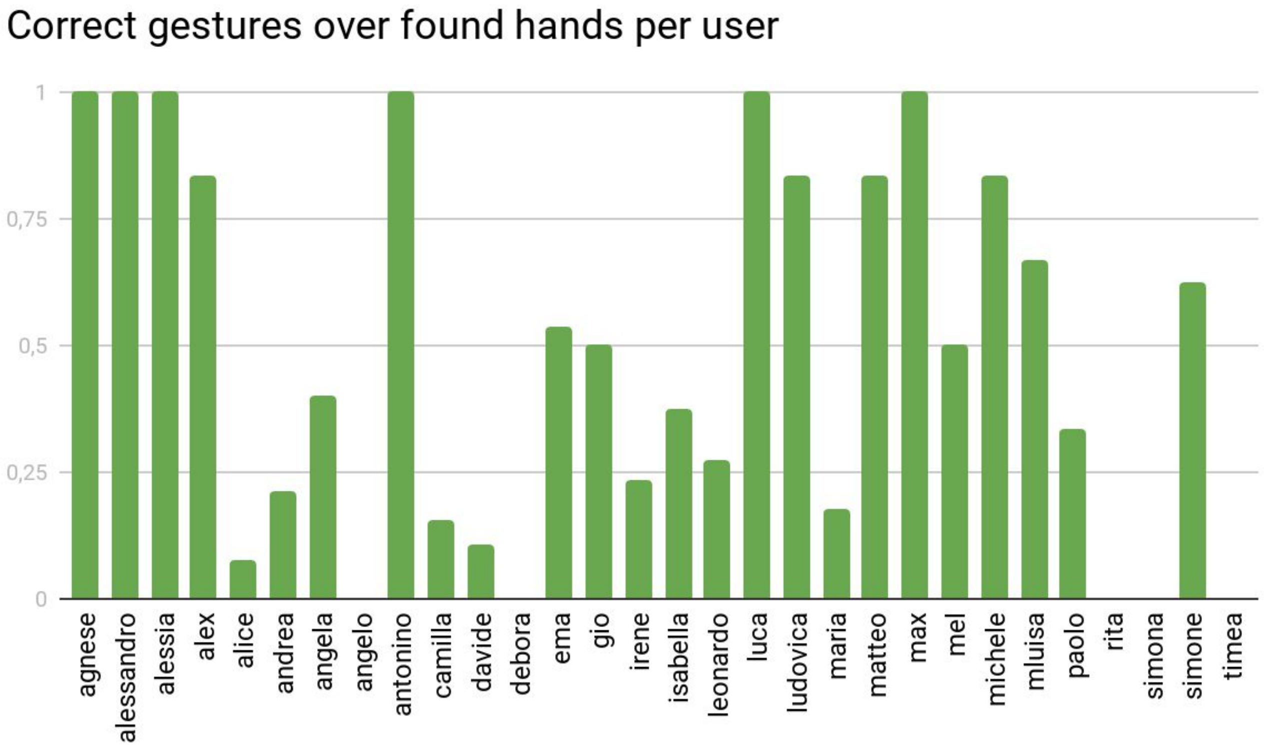
We noticed that the results are strictly user-dependent:

- cluttered or contrasting background,
- visible user or only hand,
- camera quality.





Evaluation - Gesture Recognition



Evaluation - Voice Recognition

We performed tests on all the 31 available registration (310 utterances) and 1 user out of 31 spoken very low, so the system was not able to understand the voice commands.

Tests were performed in real-time and we got:

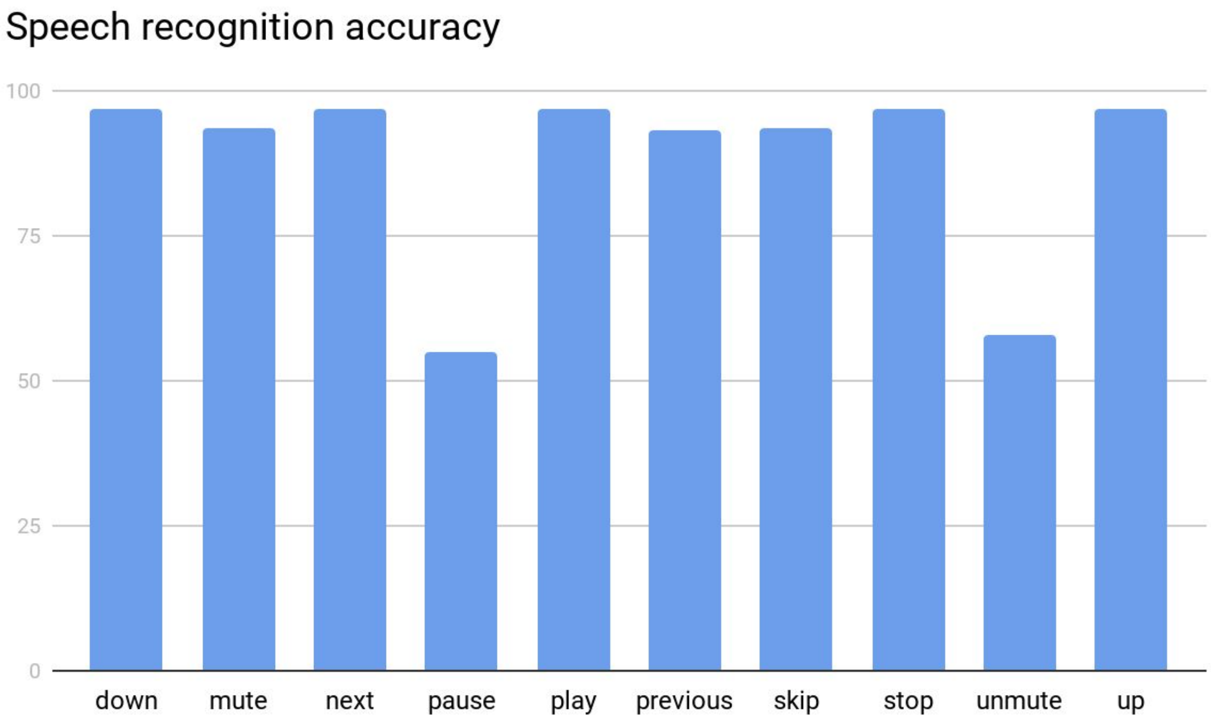
- *Accuracy* = correct_actions/n_utterances = 252/310 = **81.29%**
- *Recognition accuracy* = correct_utterances/n_utterances = 247/310 = **79.68%**

Looking at the single scores we notice that:

- **Next**, **play** and **stop** are always recognized, except for one case: **96.77%**,
- **Pause** was the utterance that was more hard to recognize: **54.84%**.

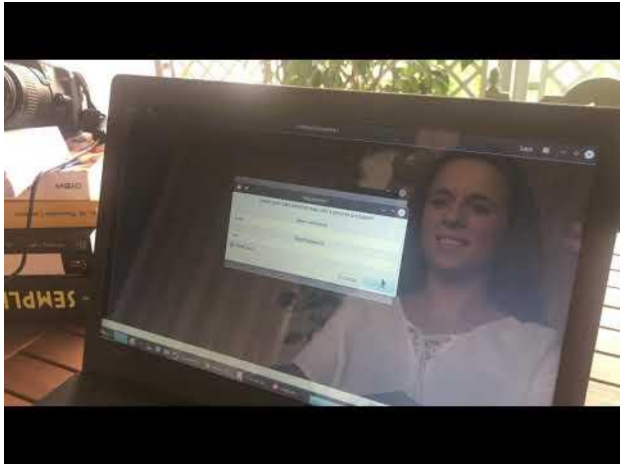
This last utterance was often recognized as *pose*, *post, bulls, those* and others.

Evaluation - Voice Recognition



Demo Deaf User

- The demo is available at the following link: https://youtu.be/1lTGk85tQ-8
- In details: the user is not already enrolled.
  1. The user performs the enrollment with the username “irene” and checks the “deaf” option.
  2. The system correctly recognized the mood of the user as neutral (the correct one is the first taken picture).
  3. The user try the following commands:
     a. Gesture “volume up” -> not recognized -> ignored
     b. Gesture “pause” -> correctly recognized
     c. Gesture “volume down” -> correctly recognized
     d. Gesture “play” -> correctly recognized
     e. Gesture “next” -> correctly recognized
     f. Gesture “volume down” -> wrongly recognized as mute -> mute
     g. Gesture “unmute” -> correctly recognized



Demo User (Not Deaf)

- The demo is available at the following link: https://youtu.be/yjkMG3RedNA
- In details: the user is already enrolled in the database as “giovanni”.
    1. The system automatically logs in the user with the correct identity.
    2. The system correctly recognized the mood of the user as happy.
    3. The user tried the following commands:
        a. Gesture “pause” -> correctly recognized
        b. Utterance “play” -> correctly recognized
        c. Utterance “skip” -> correctly recognized
        d. Gesture “previous” -> correctly recognized
        e. Utterance “next” -> correctly recognized
        f. Utterance “mute” -> correctly recognized
        g. Gesture “unmute” -> correctly recognized
        h. Utterance “stop” -> correctly recognized

Conclusions

The application could be improved in some aspects:

- **More intuitive gestures**: use a custom and more complex model in order to recognize more intuitive gestures that better represents the commands that are performed on the player;
- **Usability**: could be an interest point to carry out a survey to understand if the more intuitive gestures are good for the mute community too, and ask them to real-time try the application;
- **More responsiveness**: for which concern the gesture interaction task the application is a little bit slow.

In general we are satisfied with the results since the application satisfies its requirements.

References

- YouTubePlayer: https://github.com/vn-ki/YoutubePlayer
- GTK toolkit: https://www.gtk.org
- youtube-dl: https://github.com/ytdl-org/youtube-dl
- openCV: https://opencv.org
- gphoto2: http://www.gphoto.org
- PostgreSQL: https://www.postgresql.org
- Face++ (Gesture Recognition): https://www.faceplusplus.com/gesture-recognition/
- Face++ (Face Detection): https://www.faceplusplus.com/face-detection/
- Face++ (Face Searching): https://www.faceplusplus.com/face-searching/
- Microsoft Azure Speech To Text: https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/