Basically, the shift towards voice applications was the main driving force of this project which consists of the implementation of an in-car voice assistant using mainly a Raspberry Pi and Alexa Voice Service.

This project can be divided to two main parts: developing some Alexa skills ,like moving forward ,right, backward,turning light on...and a real time speech enhancement process to denoise commands before being received by the Alexa recognizer.

We should be able to say a command like “Alexa trigger forward”to move the robot forward.

Alexa will alert IFTTT(IF This Then That), which will send an HTTP request through ngrok to the robot and post to the web server running in Flask. The Flask program will command the robot to move forward.

* **The script of this part is located in the folder :robot\_controlling.**

In fact, due to the distance between the speaker and the assistant, the latter picks up all ambient noise and reverberation added to the main voice which are sources of interference. Moreover, there are potential competing speakers who will cause a particularly difficult understanding of the main speaker.

To facilitate understanding for this assistant, the strategy adopted is to denoise voice, before being processed by the interaction model of Alexa, by enhancing the source of interest, and thus send to the assistant a signal that is easier to interpret without ambiguities and which will be transmitted to the IFTTT service which is linked with ngrok.

To do so, we have chosen the U-Net model which is based on the convolutional neural network (CNN) to process the spectrograms resulting from the noisy requests.

* **The script of this part is located in the folder :’script’ which contains 3 python files in google colab:**
* **prepare\_data**: blend noise with clean speech and save the spectrograms of noisy voices, noises and clean voices to the drive as well as complex phases, time series and sounds in the ‘data’ folder.
* **training\_U-Net**: train the model to predict the noise from a noisy input sound using the U-Net architecture.
* **Testing\_model**:Test the results with real commands.

PS: Only basic functions have been presented in this recap video.