Final Project
CIVIL-459
Spring 2022 Final Project

EPFL
School of Civil Engineering

EPFL Tandem Race 2022

1 V100 setup

For your model to run smoothly during the race, we will provide you access to powerful V100 GPUs. It is required that you setup the environment needed for your code to work and for Loomo to communicate with your code. Follow these steps to have a working pipeline. Steps 2-6 can also work on your laptop (MAC or Linux).

Step-by-step guide

1. Login to the V100 server using your group number and the password: 123456 (you will be required to set a new password once you login):

   ```
   ssh -AY group[GROUP NUMBER]@128.178.240.162
   ```

2. Clone your code to a specific directory.

3. Set-up the virtual python environment that will contain all your required packages (It is recommended to create the virtual environment inside your project folder `$PROJECT_FOLDER`):

   (a) `virtualenv $PROJECT_FOLDER/venv`

   (b) `source $PROJECT_FOLDER/venv/bin/activate`

   (c) `pip install [YOUR PACKAGES]`

   Make sure that every time you want to run your code, you have to run the second command (b) to activate your virtual environment with all the packages.

4. Clone the DLAV repository that will have the script which will communicate with Loomo.

   ```
   git clone git@github.com:vita-epfl/DLAV-2022.git
   ```

5. The file that receives the image from Loomo and sends Loomo the bounding box of the person of interest is *client.py*. **Do not modify this script. Check it to understand how it works.**.

6. Modify the *detector.py* file to load your model and forward the image to it. The *forward* method in *detector.py* should return two items: (1) bounding box array

$[center_x, center_y, width, height]$, (2) object label. **Check detector.py before modifying it to better understand the different components and then change it to work for you.**

At this point your code should be working. You can verify that by running your model on a single image and make sure it runs. Since the server has only 4 32GB V100 installed, you will need to make sure that your code will be using only one GPU. A single GPU is enough and every group has one GPU during the race. To make sure your code runs on one GPU, you can pass the variable `CUDA_VISIBLE_DEVICES=<GPU-NUMBER >` before calling the python script, as such:

```
CUDA_VISIBLE_DEVICES=0 python client.py --ip-address <ROBOT-IP> --checkpoint
↪  <MODEL>
```

To obtain the `<ROBOT-IP>`, check the next section. To check which GPU is not being used, run the following command ***nvidia-smi*** and check the running processes (Fig. 1).

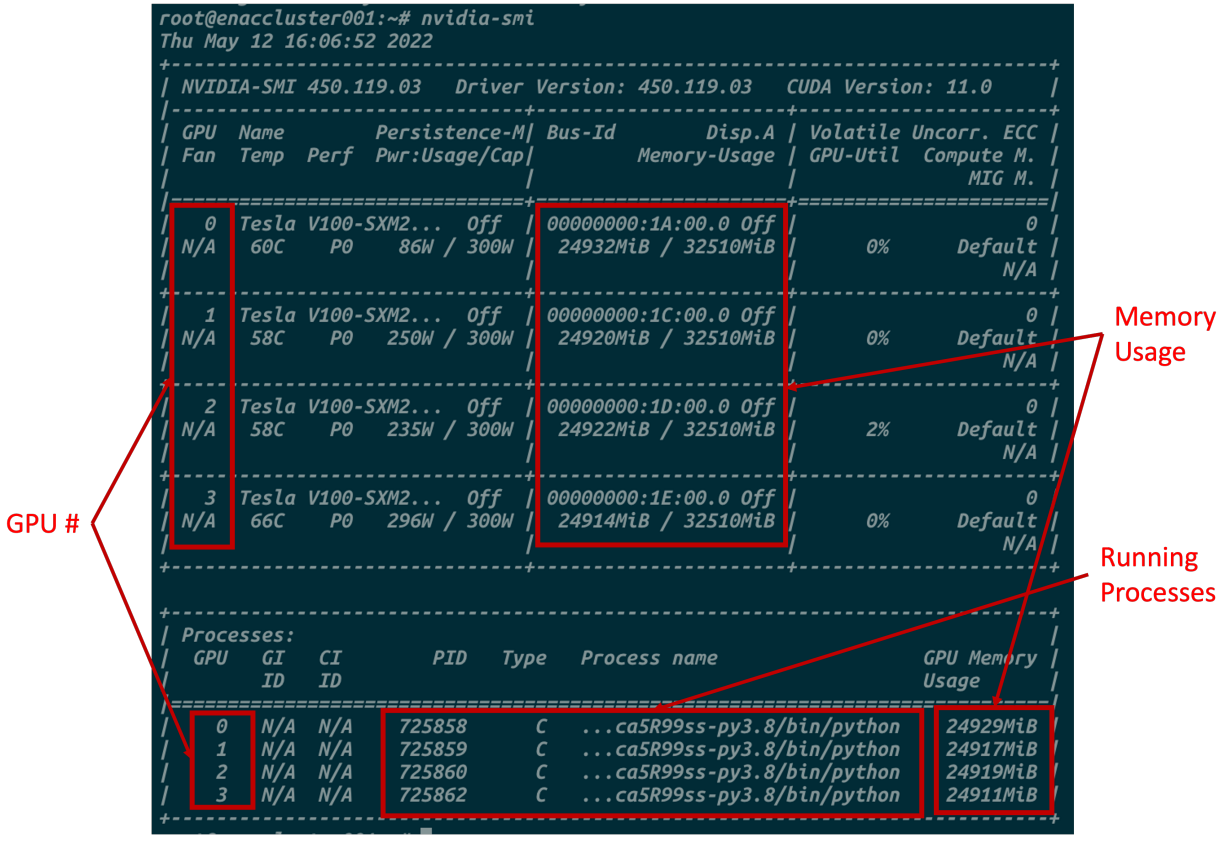
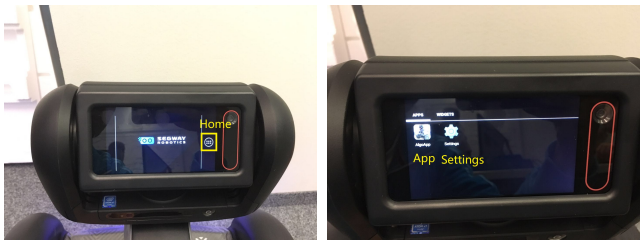


Figure 1: ***nvidia-smi*** output

2 Robot

2.1 Turn on your robot

To turn on a Loomo, press the button on its body and the button on its head in turn. Figure 2 shows the main pages after you turn on the robot.

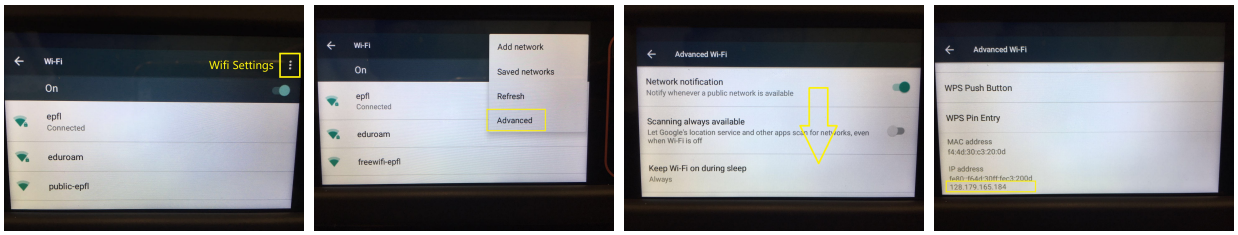


(a) Home **(b)** Apps

Figure 2: Turn on robots

2.2 Find dynamic IP address

To establish the communication between the robot and the cloud via Wi-Fi, you need to find the IP address, as shown in Figure 3.

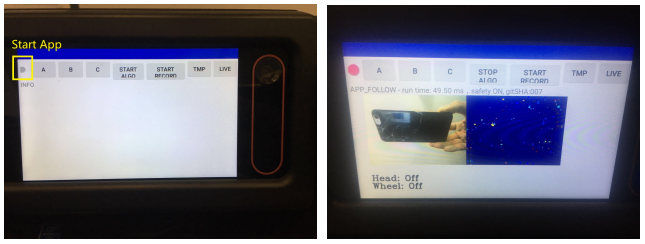


(a) Step 1 **(b)** Step 2 **(c)** Step 3 **(d)** Step 4

Figure 3: Find IP address

2.3 Start robot app

You can now get back to the main page by tapping the robot's ear. Figure 4 shows how to start the robot app for person tracking.



(a) Before Start **(b)** After Start

Figure 4: Start App

2.4 Turn on controllers

Once the app is initialized, you can switch on the controller of the robot's head and wheel with the button

- A: head control
- B: wheel control

2.5 Cloud

Once the robot is settled, you can now start your model on the cloud (your laptop or the server). The codes can be found here.

client.py file receives images from robot and passes them to *detector.py*. Then, *detector.py* file receives the images and sends back the position of the person to *client.py* which also sends back the data to the robot. **Note: You should not change *client.py*. Thus, note the changes in *detector.py* to be consistent with the *client.py* code.**

1. Test the IP address and the video stream (inside python folder)

```
python test.py --ip-address <ROBOT-IP>
```