Waveshare's JetBot

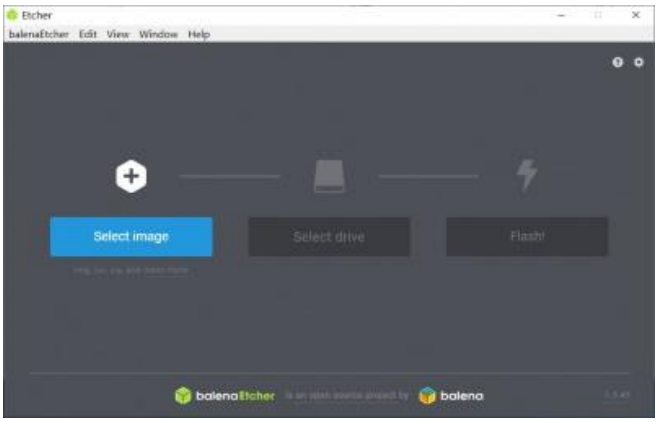
This JetBot supports the Face Recognition and Object Tracking. It uses Artificial Intelligence and Machine Learning to perform tasks. There are some basic steps are follows to create an environment for JetBot Operations.

Basic Requirements: JetBot AI kit, SD card

Steps to create JetBot AI Kit based on JetSon Nano developer kit:

Step 1: Image configuration

- You need to prepare an SD card which should be at least 64G.
- Download the JetBot image which is provided by NVIDIA and unzip it. https://drive.google.com/file/d/1o08RPDRZuDloP_o76tCoSngvq1CVuCDh/view
- Connect the SD card to the PC via a card reader.
- User Etcher software to write the image (unzip above) to the SD card.
- Download Balena Etcher https://etcher.balena.io/



- Flash image on SD card
- Remove the SD card from card reader and insert the SD card into JetBots SD card. SD card slot is on the baseboard, below the core module to the left
- Boot the JetBot

Step 2: External Peripheral devices connection

- Connect the mouse, keyboard, display to the JetBot
- Connect the USB to the laptop and plug the c code in the JetBots plug

Step 3: Clone the git repository

- Command prompt
    1. git clone http://github.com/NVIDIA-AI-IOT/jetbot.git
    2. cd jetbot
    3. cd docker

Step 4: Connect the JetBot by wireless connection Wi-Fi

- Commands on windows power shell
    1. ssh
    2. ssh jetbot@<Default IP Address> i.e 192.168.55.1
    3. Password: jetbot
    4. sudo nmcli device wifi connect Logsun password 12345678

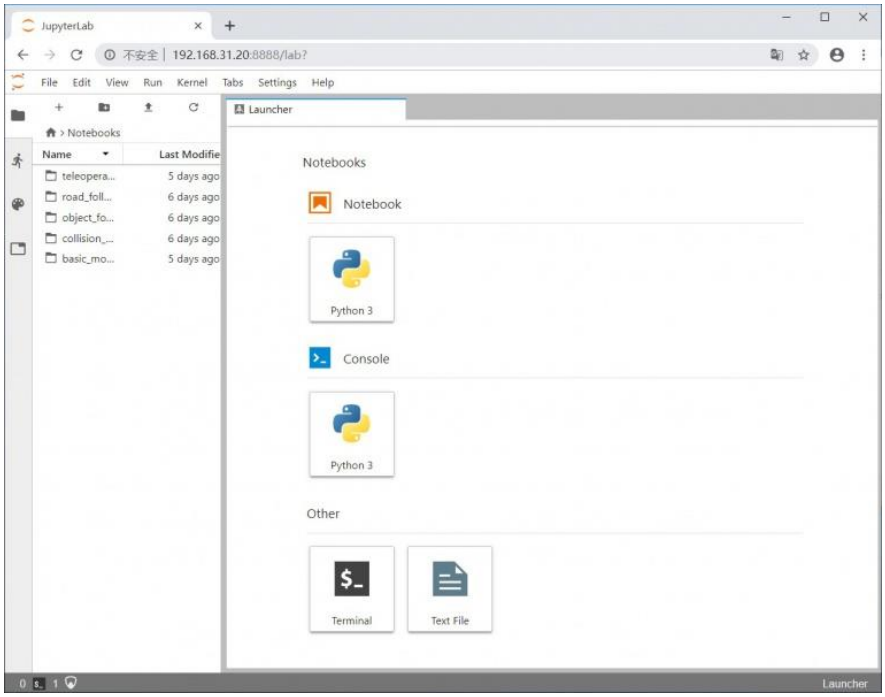
       (to check password of the wifi go to cmd=>
       netsh wlan show profile name="Logsun" key=clear)
    5. sudo shutdown now
- Power off. Assemble the Jetbot trolley and start Jetson nano. When starting up, the system will automatically connect to WIFI and display the IP address on the OLED display at the same time.
- default IP address of 192.168.55.1

Step 5: Access JetBot via Web

- After networking. You can remove peripherals and power adapter.
- Turn Power switch of Jetbot into on.
- After booting, IP address of OLED can be displayed on OLED.
- Navigate to http://<jetbot_IP_Address>:8888 from your desktop's web browser.

Step 6: Install latest software

- http://<jetbot_IP_Address>:8888 Password: Jetbot
- You will see this window if your SD card contains that git repository



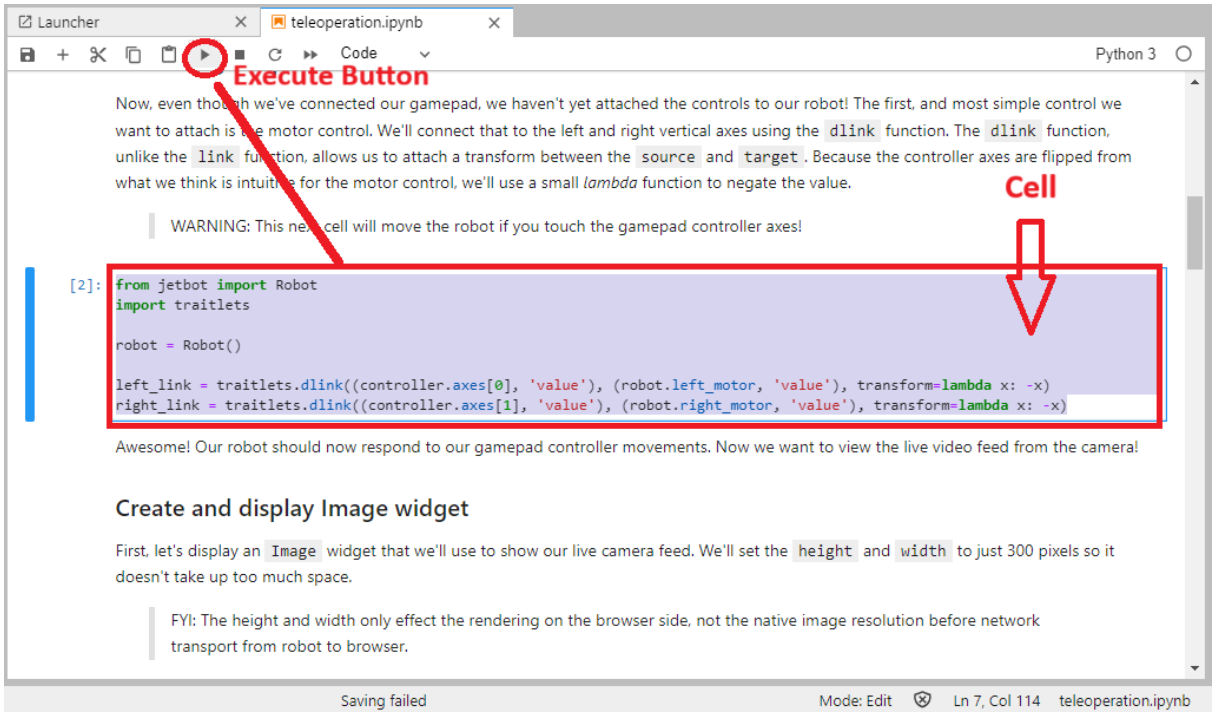
Step 7: Configure Power Mode

- The following command can be used to switch the power consumption mode of Jetson Nano to 5W mode to ensure that the battery pack can supply power normally.
- Open the browser http://<jetbot_IP_Address>:8888 to connect to the car, and start another terminal.
- Switch to 10W power mode.

    1. sudo nvpmodel –m1
    2. nvpmodel –q

Step 8: Basic motions

- Access jetbot by going to http://<jetbot_IP_Address>:8888 navigate to ~/Notebooks/basic_motion/
- Open basic_motion.ipynb file and following the notebook.
- **Note:** You can click icon ▶ to run codes, or select Run -> Run Select Cells. Make sure the JetBot has enough space to run.


Launcher
teleoperation.ipynb
Code
Python 3
Execute Button
Cell
Now, even though we've connected our gamepad, we haven't yet attached the controls to our robot! The first, and most simple control we want to attach is the motor control. We'll connect that to the left and right vertical axes using the dlink function. The dlink function, unlike the link function, allows us to attach a transform between the source and target. Because the controller axes are flipped from what we think is intuitive for the motor control, we'll use a small lambda function to negate the value.
WARNING: This next cell will move the robot if you touch the gamepad controller axes!
[2]:
from jetbot import Robot
import traitlets
robot = Robot()
left_link = traitlets.dlink((controller.axes[0], 'value'), (robot.left_motor, 'value'), transform=lambda x: -x)
right_link = traitlets.dlink((controller.axes[1], 'value'), (robot.right_motor, 'value'), transform=lambda x: -x)
Awesome! Our robot should now respond to our gamepad controller movements. Now we want to view the live video feed from the camera!
Create and display Image widget
First, let's display an Image widget that we'll use to show our live camera feed. We'll set the height and width to just 300 pixels so it doesn't take up too much space.
FYI: The height and width only effect the rendering on the browser side, not the native image resolution before network transport from robot to browser.
Saving failed
Mode: Edit
Ln 7, Col 114
teleoperation.ipynb