# Installation of Jupyter lab and Jetcham

Instructions before use

This tutorial is applicable to self built images. If you are using a YAHBOOM version of the image, which is already installed, you can ignore it and ignore it

Introduction to JupyterLab

The JupyterLab interface is a dashboard that provides access to interactive iPhone laptops, as well as the folder structure of Jetbots and terminal windows for accessing the Ubuntu operating system. The first view you will see includes the menu bar at the top, the directory tree in the left sidebar, and the main workspace that was originally opened to the "launcher" page.
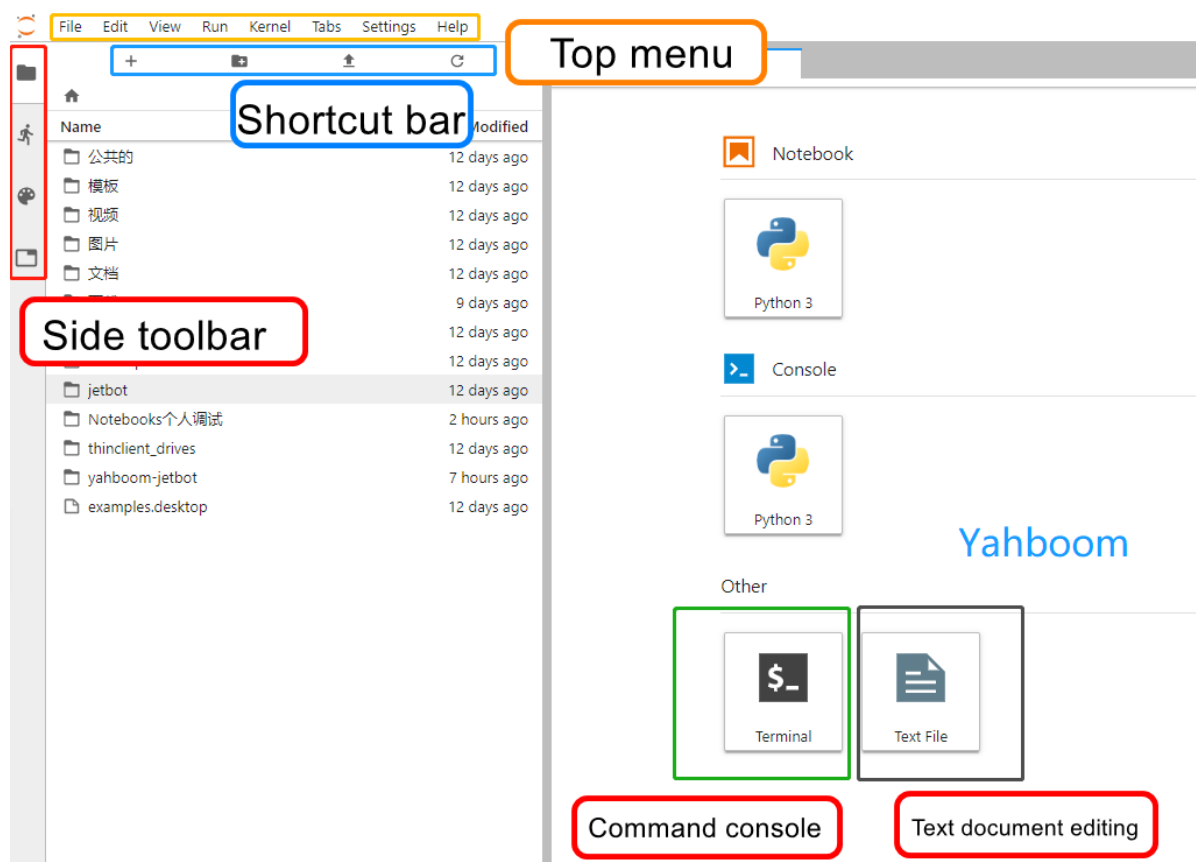
For complete and detailed information on all functions and menu operations, please refer to

https://jupyterlab.readthedocs.io/en/stable/user/interface.htmlfile. The following are some key features that are particularly useful in this course:
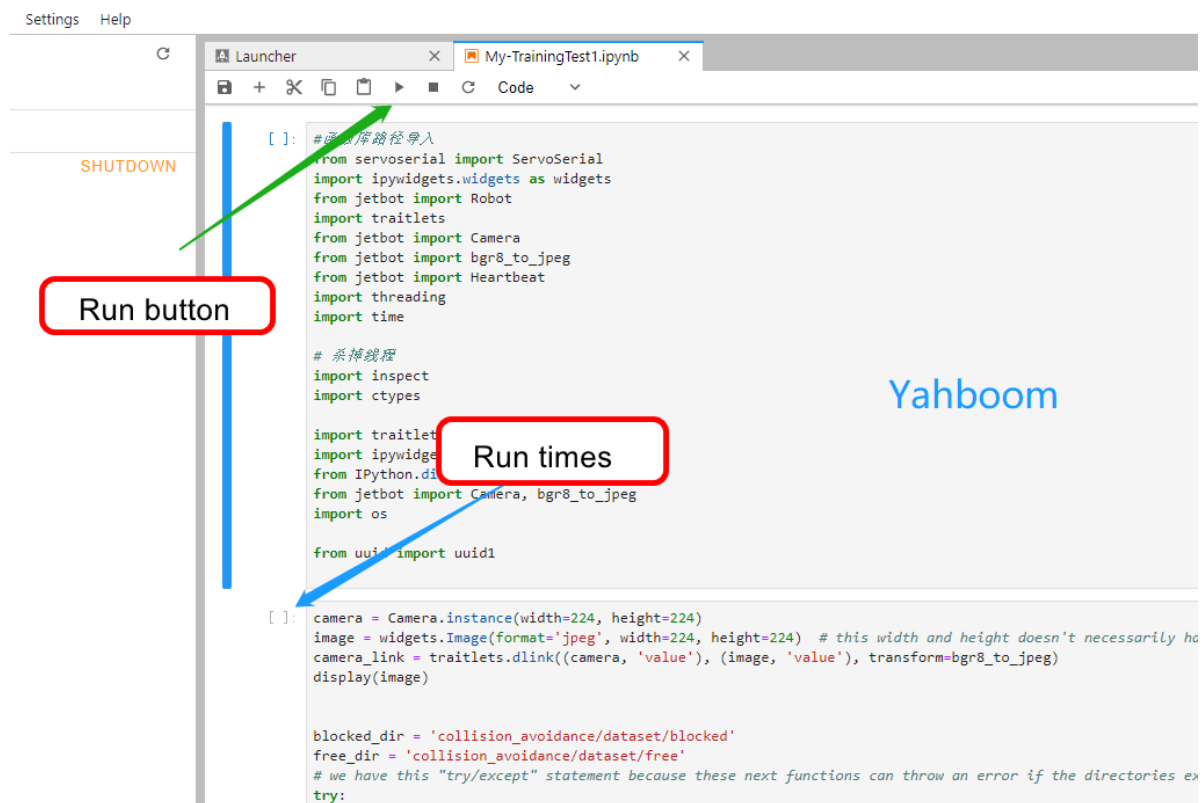
- File Browser:

The file browser in the left column allows navigation to Jetson Nano B01 file structure. Double clicking on a notebook or file will open it in the main workspace.

- IPhone laptop:

The interactive notebook used in this lesson has a ". ipynb" filename extension. When you double-click a notebook from the file browser, it will open in the main workspace and the process will begin. The notebook includes text and code 'cells'. When the code unit "runs", click the run button or keyboard shortcut [CTRL] at the top of the notebook+[ENTER] will execute the code block in the

cell and display the output (if any) below the laptop. To the left of each executable cell, there are "execution count" or "prompt number" in parentheses. If the cell runs for more than a few seconds, you will see an asterisk there indicating that the cell has not yet completed execution. After completing the processing of the cell, a number will be displayed in parentheses.


- Kernel operation:

The kernel of each running laptop is an independent process that runs user code. When opening a notebook from a file browser, the kernel will automatically start. The kernel menu on the main menu bar contains commands to shut down or restart the kernel, which you need to use regularly. After the kernel is shut down, no code units can be executed. When restarting the kernel, all memory will be lost due to imported packages, variable assignments, etc.
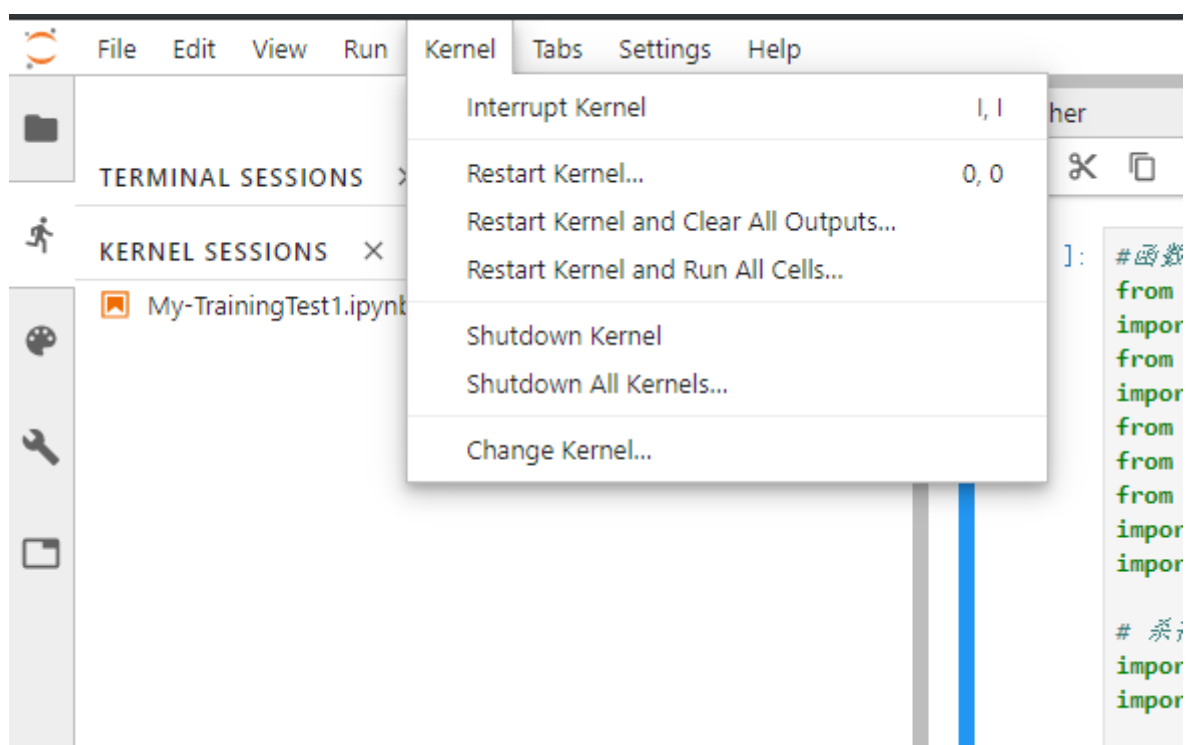
Cell labels:

You can move any cell to a new window tab in the main workspace by right-clicking on the cell and selecting 'Create New View for Output'. This way, you can continue scrolling down the JupyterLab notebook while still viewing specific cells. This is particularly useful in cells containing camera views!!
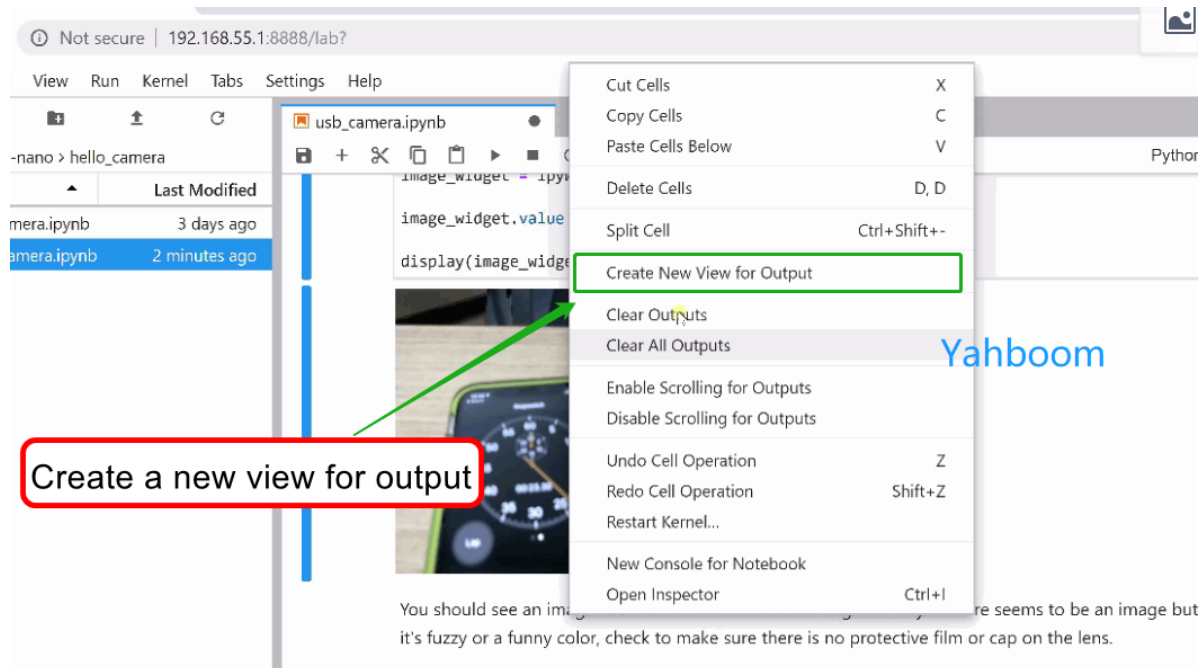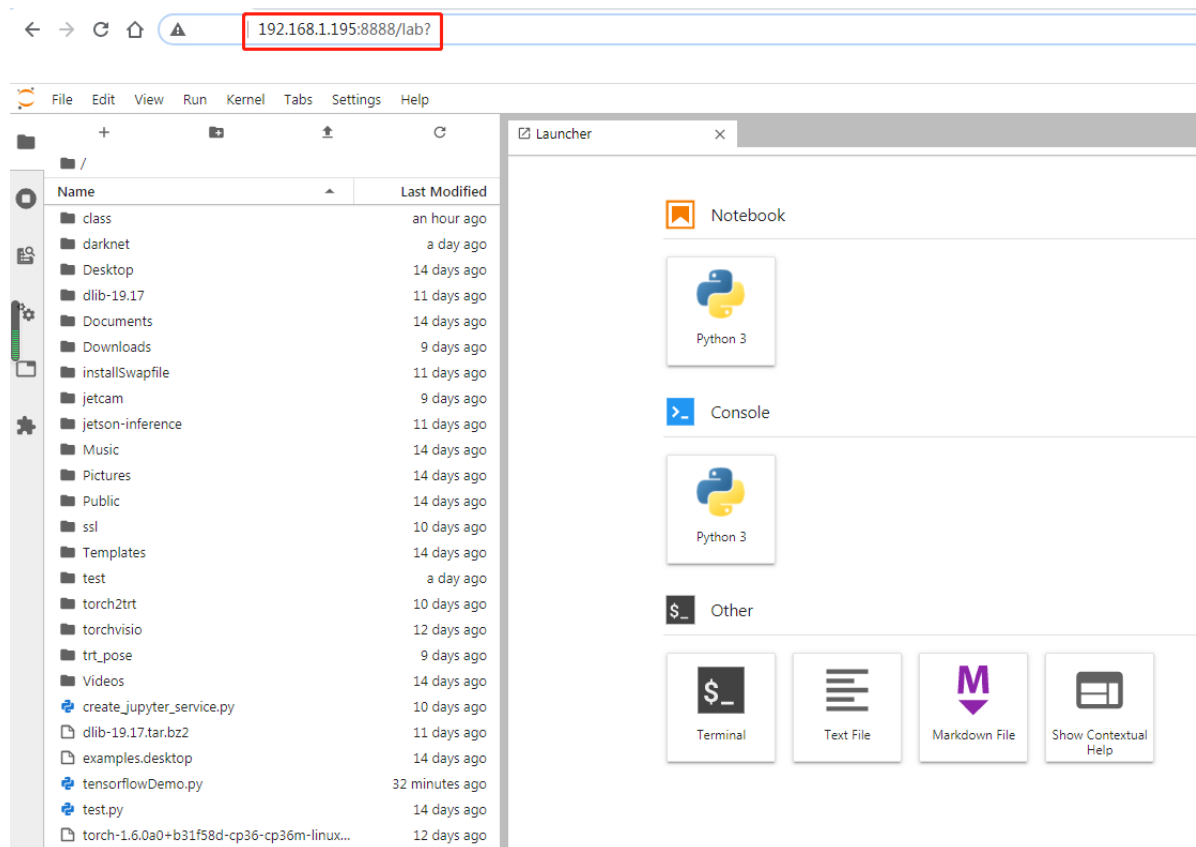

- Terminal：

You can directly log in remotely to JetbotUbuntu OS through JupyterWork in the terminal window of. On the Launcher page, click the terminal icon under 'Other'. To bring up the Launcher page, if it is no longer visible, click the "+" icon at the top of the left column.
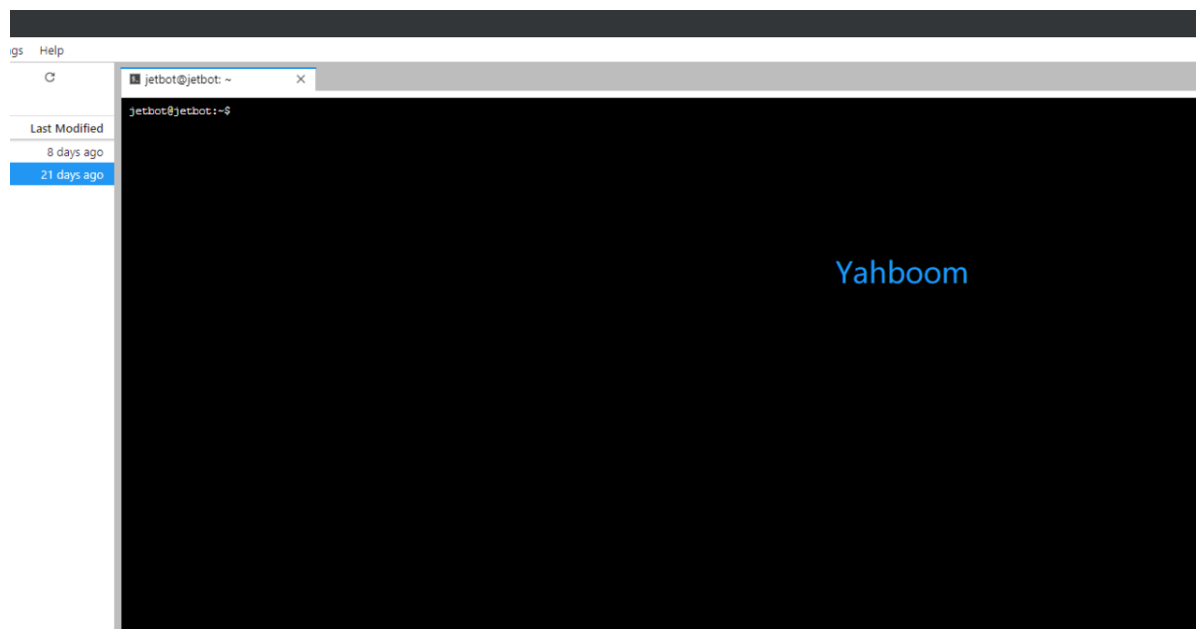
The login method for JupyterLab is to enter IP: 8888 at the input end of the browser webpage

For a brief introduction, let's hurry up and install it now (the configured image does not need to be installed, and Jumperlab has been started after booting up. You can directly log in to the webpage based on your IP and password, with the password being yahboot)

Install nodejs and npm

```
sudo apt-get update

sudo apt install nodejs npm
```

However, using the above command directly to install the version is relatively low. If you need to install the jupyterab plugin in the future, you may encounter errors. You can check with this version, and at least version 12.3.0 of the node is required

```
node -v

npm -v
```

Install the n module and use it to update or specify the version of the node to be installed

```
npm install -g n
```

First, please explain the functions of this module and follow the commands to understand it. There is no need to operate itClear npm cache: npm cache clean - fInstall n modules: npm install - g nInstall the official stable version: n stableInstall the latest official version: n latestInstall a specific version: n 11.6.0View installed node versions: nView the current node version: node - vDelete specified version: n rm 7.5.0

Okay, after understanding the functions of the n module, install the corresponding version of the node, or you can install the latest version, such as the following

```
sudo n latest
```

After installation, node - vCheck the version. If the version has not changed, you can try restarting it. If it still hasn't changed, execute

```
sudo n
```

A screen will appear, showing the version name of the node we have installed. For example, we have v15.0.1 here. Use the up and down arrow keys to control the cursor to select this version, then press Enter to mirror the installation, and then check the version. If it still hasn't changed, usually restart it again.

Install jupyteralab: (Warning ignored, multiple failed executions are sufficient)

```
sudo pip3 install jupyter jupyterlab

sudo jupyter labextension install @jupyter-widgets/jupyterlab-manager

sudo jupyter labextension install @jupyterlab/statusbar
```

Generate corresponding configuration files: (If a file reports permission issues, you can try using sudo chmod)777 granting permissions)

```
jupyter lab --generate-config
```

Set the password to enter the notebook (here it will be set twice, the second time to confirm the password entered):

```
jupyter notebook password
```

When logging in and using Notebook for the first time, you need to enter the password you set here to enter. Please remember the current password set!Set up bootable jumperlab, create_ jupyter_ The service.py file is located in the attachment file of the environment configurationRun create_ jupyter_ Service. py file generates jupyter_ Service.service file

```
pyhon3 create_jupyter_service.py
```

Then move the generated service file to the system service

```
sudo mv nano_jupyter.service /etc/systemd/system/nano_jupyter.service
# Enable this service
sudo systemctl enable nano_jupyter.service
# Manually activate the service
sudo systemctl start nano_jupyter.service
```

Installing Jetcam

JetCam is used for NVIDIA Jetson's easy-to-use Python camera interface. Using Jetson's AcceleratedThe GStreamer plugin can be used in conjunction with various USB and CSI cameras. Easy to read images as numpy array images= camera.read()。 Set camera to run=True appends the callback to the new framework. JetCam makes it easy to create prototypes for AI projects in Python, especially Jupyter installed in JetCardIn the Lab programming environment.

Next, start installing:

```
git clone <https://github.com/NVIDIA-AI-IOT/jetcam>

cd jetcam

sudo python3 setup.py install
```

Detailed usage of the function can be foundhttps://github.com/NVIDIA-AI-IOT/jetcam