Establishment of Tensorflow Lite and OpenVINO IR model runtime environment

Hardware: Raspberry Pi 3/4 and Intel® Neural Compute Stick 2

Simulation Raspberry Pi 3B

Author: Yixiao Yuan

Apr 10,2023

Establishment of TF Lite and IR model runtime environment

1. Raspbian OS

Hardware requirements: Raspberry Pi 3 or 4 and Intel® Neural Compute Stick 2

Operating system: Raspbian Buster, ARM, 32-bit or Raspbian Stretch, ARM, 32-bit

Raspberry Pi OS Lite : raspios_lite_armhf-2021-05-28

2. Install Tensorflow lite

```
sudo apt install python3-pip

echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable main" | sudo
tee /etc/apt/sources.list.d/coral-edgetpu.list
curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

sudo apt-get update
sudo apt-get install python3-tflite-runtime

pip3 install psutil
```

Use the `sudo pip3 list` command to verify that the installed packages include `numpy` and `tflite-runtime`, and then you can successfully run the tflite model.

3. Install OpenVINO™ Runtime for Raspbian OS

Reference: https://docs.openvino.ai/2022.2/openvino_docs_install_guides_overview.html

3.1 Download and Install OpenVINO Runtime

Create an installation folder for OpenVINO. If the folder already exists, skip this step.

```
sudo mkdir -p /opt/intel
```

Go to your ~/Downloads directory and download OpenVINO Runtime archive file

```
mkdir ~/Downloads/
cd ~/Downloads/

sudo wget
https://storage.openvinotoolkit.org/repositories/openvino/packages/2022.2/linux/l_op
envino_toolkit_debian9_arm_2022.2.0.7713.af16ea1d79a_armhf.tgz -O
openvino_2022.2.0.7713.tgz
```

Extract the archive file and move it to the installation folder:

```
sudo tar -xf openvino_2022.2.0.7713.tgz

sudo mv l_openvino_toolkit_debian9_arm_2022.2.0.7713.af16ea1d79a_armhf
/opt/intel/openvino_2022.2
```

Install required system dependencies on Linux.

```
cd /opt/intel

sudo ln -s openvino_2022.2 openvino_2022

sudo apt install cmake
```

3.2 Configurations for Intel® Neural Compute Stick 2

Add the current Linux user to the users group:

```
sudo usermod -a -G users "$(whoami)"
```

If you didn't modify `.bashrc` to permanently set the environment variables, run `setupvars.sh` again after logging in:

```
source /opt/intel/openvino_2022/setupvars.sh
```

To perform inference on the Intel® Neural Compute Stick 2, install the USB rules running the `install_NCS_udev_rules.sh` script:

```
cd /opt/intel/openvino_2022/install_dependencies

./install_NCS_udev_rules.sh
```

Plug in your Intel® Neural Compute Stick 2.

3.3 Set the Environment Variables

```
cd ~

source /opt/intel/openvino_2022/setupvars.sh
```

The above command (`source /opt/intel/openvino_2022/setupvars.sh`)must be re-run every time you start a new terminal session. you will see `[setupvars.sh] OpenVINO environment initialized.`, and here we can happily run the OpenVINO IR model.

To set up Linux to automatically run the command every time a new terminal is opened, open `~/.bashrc` in your favorite editor and add `source /opt/intel/openvino_2022/setupvars.sh` after the last line. Next time when you open a terminal, you will see `[setupvars.sh] OpenVINO™ environment initialized`. Changing `.bashrc` is not recommended when you have multiple OpenVINO versions on your machine and want to switch among them.

3.4. Test runtime environment and drivers

Restart the system and execute `source /opt/intel/openvino_2022/setupvars.sh`, then proceed with the following Python code:

```python
import openvino.inference_engine as IECore

engine = IECore.IECore()

print(engine.available_devices)
```

The output is: ['CPU', 'MYRIAD'], indicating that the runtime environment and the corresponding driver have been successfully installed.

Simulation Raspberry Pi 3B

1. Install QEMU.

QEMU is an open-source emulator and virtualizer that allows users to simulate various hardware architectures on a host machine. It is widely used for testing and development purposes, as well as for running legacy software or operating systems on modern hardware. QEMU can simulate a wide range of architectures, including x86, ARM, MIPS, PowerPC, and more. It can also emulate various devices, such as graphics cards, network cards, and USB devices, making it a versatile tool for developers and researchers.

QEMU is known for its high performance and reliability, and it has been used in various industries, such as aerospace, automotive, and telecommunications. It supports various operating systems, including Linux, Windows, and macOS, and it can run on various hardware platforms, including x86, ARM, and PowerPC.

QEMU is licensed under the GNU General Public License (GPL) version 2 or later, which means that it is free and open-source software. This allows users to modify and distribute QEMU as they see fit, making it a popular choice for hobbyists, researchers, and open-source developers.

To learn more about QEMU, you can visit the official website at https://www.qemu.org/.

Installation steps for QEMU 7.2.0:

1. Download https://qemu.weilnetz.de/w64/qemu-w64-setup-20221230.exe
2. Install QEMU 7.2.0 and Add the QEMU installation path to the system path.

2. Use QEMU to simulate the Raspberry Pi 3B

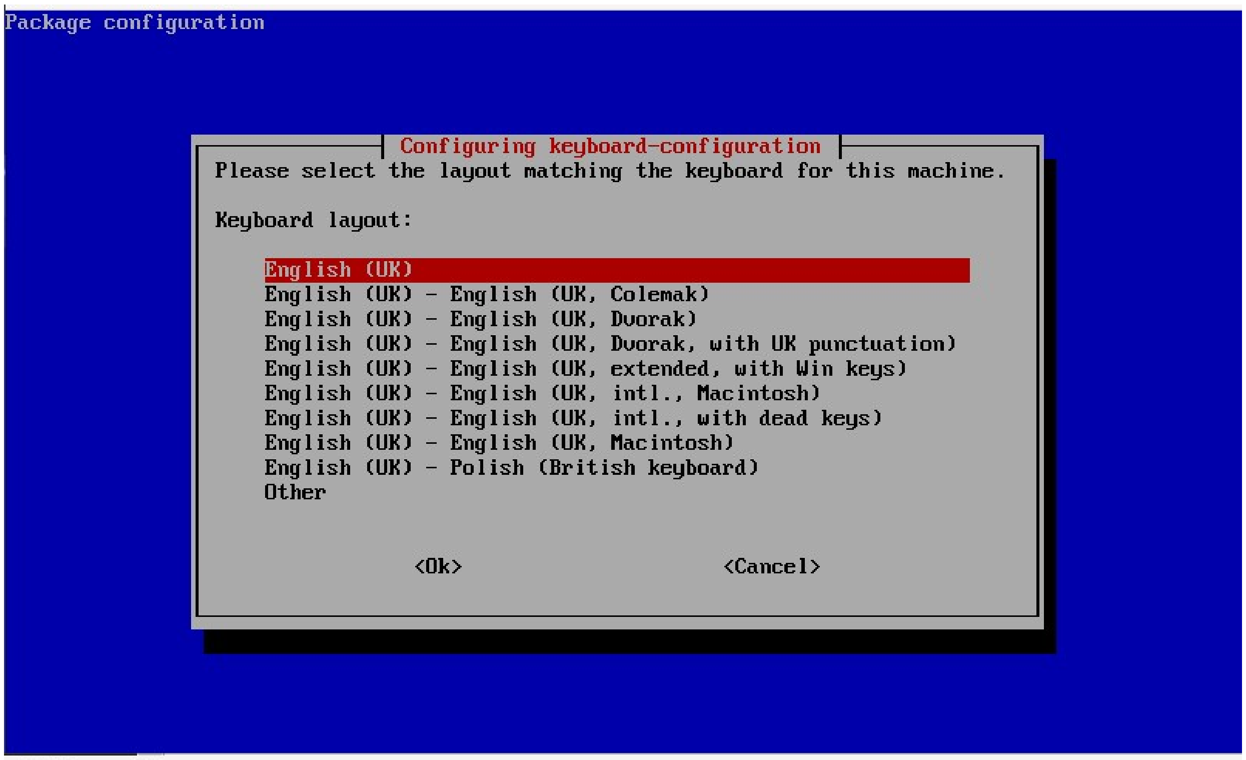
1. Download 32-bit Raspberry Pi OS Lite from the official website : raspios_lite_armhf-2021-05-28
2. Use 7zip to extract the `2021-05-07-raspios-buster-armhf-lite.zip` file, which will result in the `2021-05-07-raspios-buster-armhf-lite.img` file.
3. Use 7zip again to extract the `2021-05-07-raspios-buster-armhf-lite.img` file, which will generate two files, `1.img` and `0.fat`.
4. Use 7zip once more to extract the `0.fat` file.
5. Copy the following files, `2021-05-07-raspios-buster-armhf-lite.img`, `kernel8.img`, and `bcm2710-rpi-3-b.dtb`, into the same directory.
6. Use the following command to modify the size of the Raspberry Pi img file size to 8G, which is equivalent to using an 8G TF card when emulating the system.

```
qemu-img resize -f raw 2021-05-07-raspios-buster-armhf-lite.img 8G
```

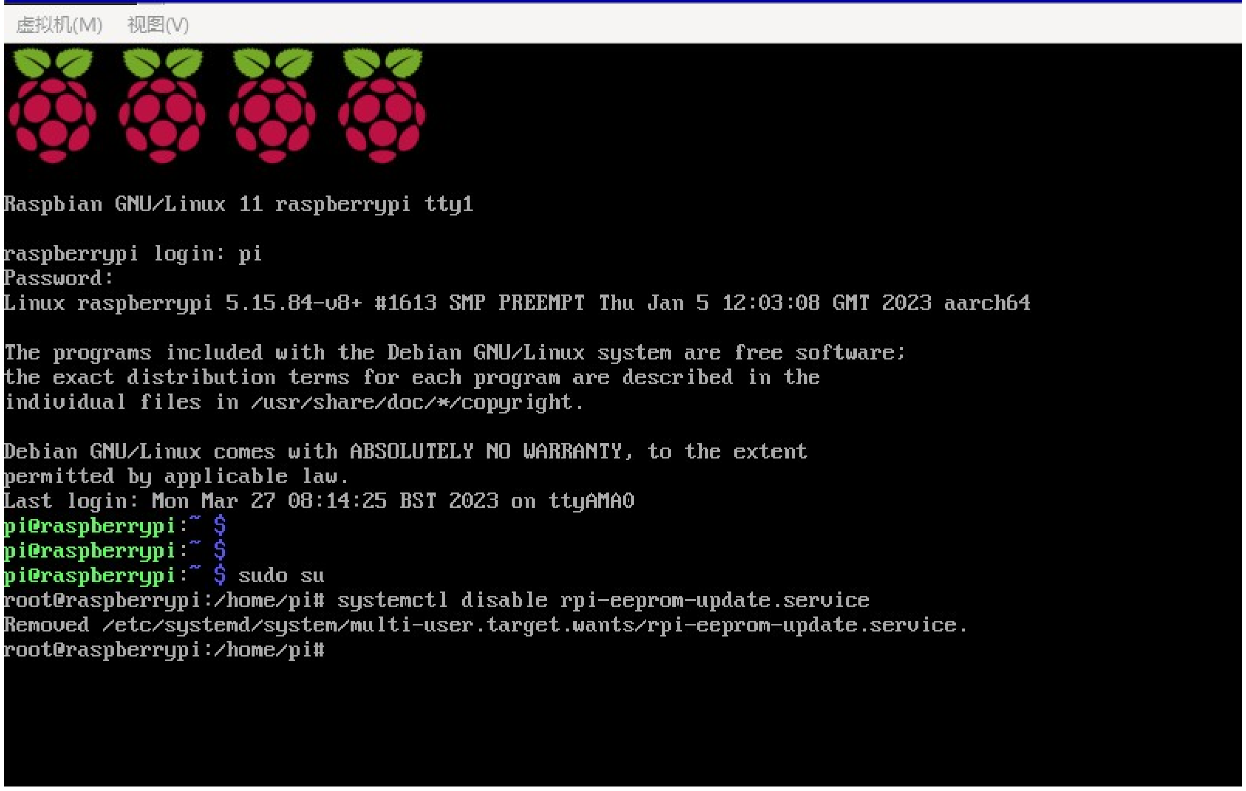
7. Boot Raspberry Pi image files with qemu-system-aarch64 command.

```
qemu-system-aarch64 -M raspi3b -append "rw earlyprintk loglevel=8
console=ttyAMA0,115200 dwc_otg.lpm_enable=0 root=/dev/mmcblk0p2 rootdelay=1" -dtb
bcm2710-rpi-3-b.dtb -drive file=2021-05-07-raspios-buster-armhf-lite.img,format=raw
-kernel kernel8.img -m 1G -smp 4 -serial stdio -netdev
user,id=net0,hostfwd=tcp::5555-:22  -usb -device usb-kbd -device usb-tablet  -device
usb-net,netdev=net0
```

8. After startup, the system setup window appears, set the user name and password to log in to the system.



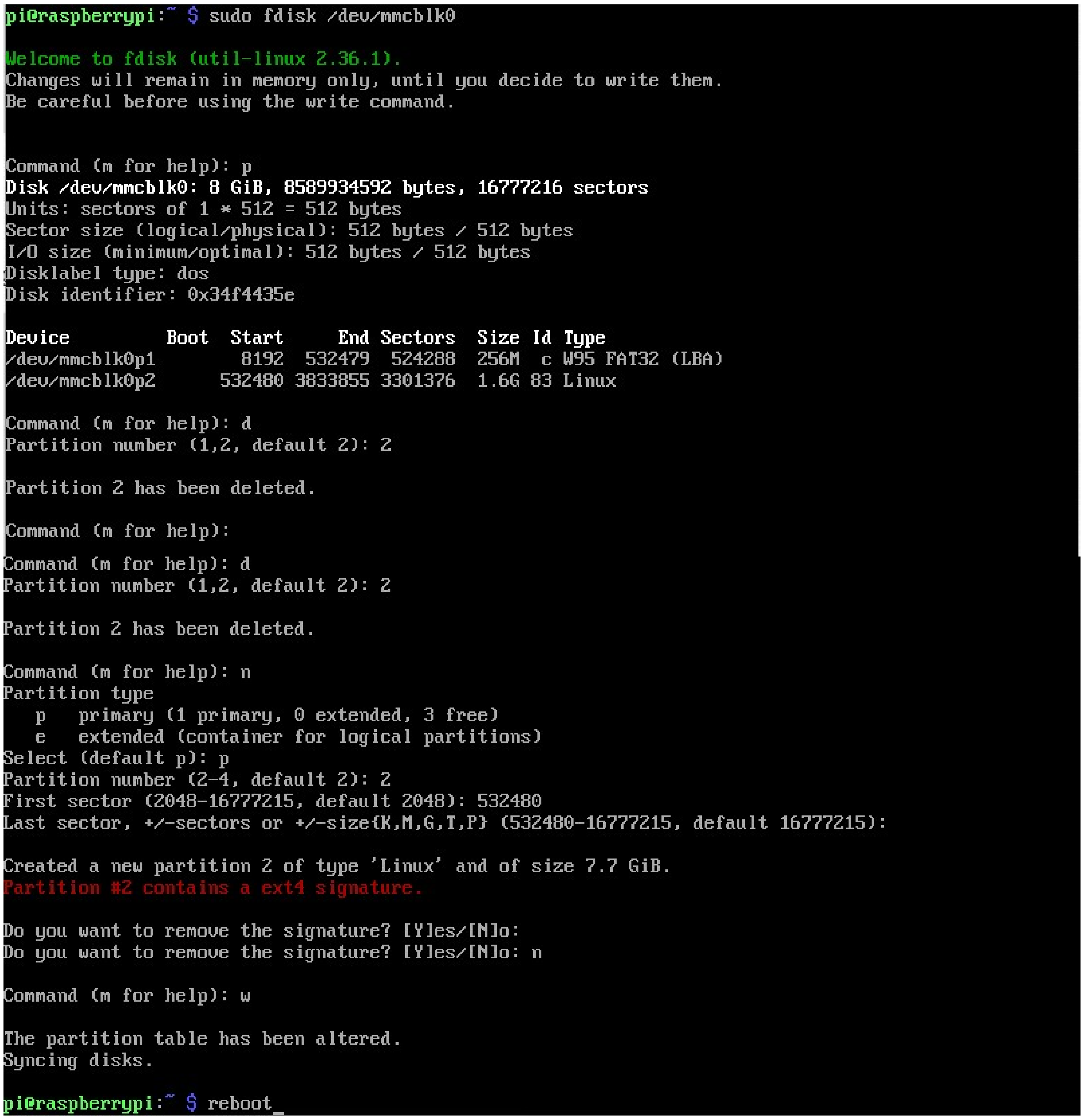
```
Package configuration

┤ Configuring keyboard-configuration ├
Please select the layout matching the keyboard for this machine.

Keyboard layout:

    English (UK)
    English (UK) - English (UK, Colemak)
    English (UK) - English (UK, Dvorak)
    English (UK) - English (UK, Dvorak, with UK punctuation)
    English (UK) - English (UK, extended, with Win keys)
    English (UK) - English (UK, intl., Macintosh)
    English (UK) - English (UK, intl., with dead keys)
    English (UK) - English (UK, Macintosh)
    English (UK) - Polish (British keyboard)
    Other

              <Ok>                        <Cancel>
```



```
Raspbian GNU/Linux 11 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 5.15.84-v8+ #1613 SMP PREEMPT Thu Jan 5 12:03:08 GMT 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 27 08:14:25 BST 2023 on ttyAMA0
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# systemctl disable rpi-eeprom-update.service
Removed /etc/systemd/system/multi-user.target.wants/rpi-eeprom-update.service.
root@raspberrypi:/home/pi#
```

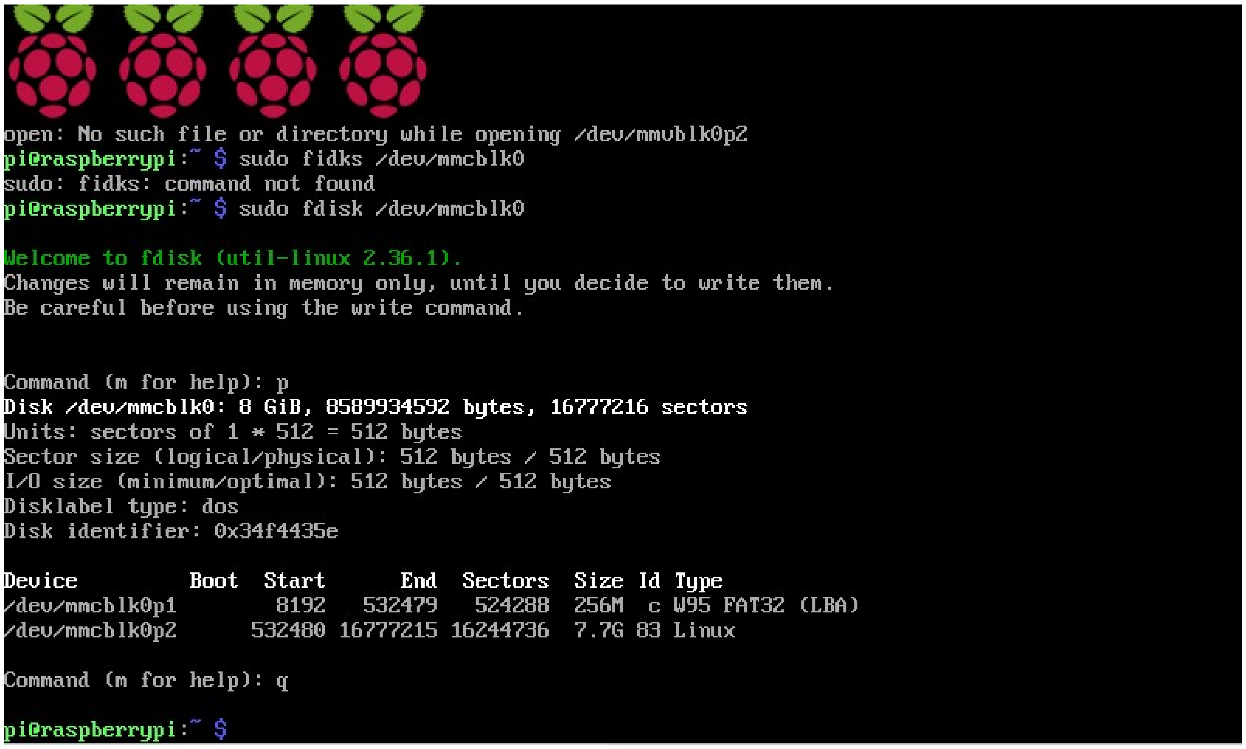
9. Raspberry Pi extended root root partition

```
sudo fdisk /dev/mmcblk0
```



```
pi@raspberrypi:~ $ sudo fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/mmcblk0: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x34f4435e

Device         Boot  Start     End Sectors  Size Id Type
/dev/mmcblk0p1        8192  532479  524288  256M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      532480 3833855 3301376  1.6G 83 Linux

Command (m for help): d
Partition number (1,2, default 2): 2

Partition 2 has been deleted.

Command (m for help):
Command (m for help): d
Partition number (1,2, default 2): 2

Partition 2 has been deleted.

Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): 2
First sector (2048-16777215, default 2048): 532480
Last sector, +/-sectors or +/-size{K,M,G,T,P} (532480-16777215, default 16777215):

Created a new partition 2 of type 'Linux' and of size 7.7 GiB.
Partition #2 contains a ext4 signature.

Do you want to remove the signature? [Y]es/[N]o:
Do you want to remove the signature? [Y]es/[N]o: n

Command (m for help): w

The partition table has been altered.
Syncing disks.

pi@raspberrypi:~ $ reboot_
```

```
sudo resize2fs /dev/mmcblk0p2
```

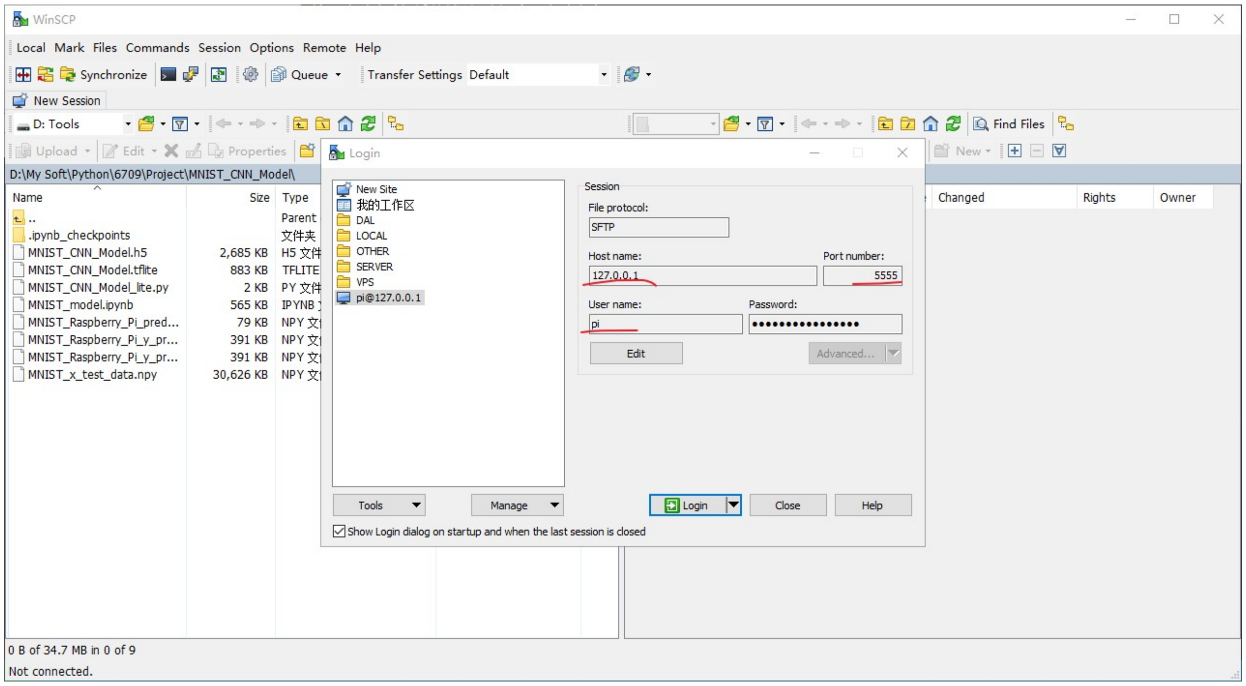


```
open: No such file or directory while opening /dev/mmvblk0p2
pi@raspberrypi:~ $ sudo fidks /dev/mmcblk0
sudo: fidks: command not found
pi@raspberrypi:~ $ sudo fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/mmcblk0: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x34f4435e

Device          Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1         8192   532479   524288  256M  c W95 FAT32 (LBA)
/dev/mmcblk0p2       532480 16777215 16244736  7.7G 83 Linux

Command (m for help): q

pi@raspberrypi:~ $
```

9. The Raspberry Pi SSH service is automatically started on boot, You can reinitialize ssh with the following command:

```
sudo systemctl enable ssh
sudo systemctl start ssh
sudo reboot
```

Use ssh client software, such as winscp, to configure the ip, port, username and password, and then you can upload and download files.



D:\My Soft\Python\6709\Project\MNIST_CNN_Model\

Name	Size	Type
..		Parent
.ipynb_checkpoints		文件夹
MNIST_CNN_Model.h5	2,685 KB	H5 文件
MNIST_CNN_Model.tflite	883 KB	TFLITE
MNIST_CNN_Model_lite.py	2 KB	PY 文件
MNIST_model.ipynb	565 KB	IPYNB
MNIST_Raspberry_Pi_pred...	79 KB	NPY 文
MNIST_Raspberry_Pi_y_pr...	391 KB	NPY 文
MNIST_Raspberry_Pi_y_pr...	391 KB	NPY 文
MNIST_x_test_data.npy	30,626 KB	NPY 文

Session
File protocol: SFTP
Host name: 127.0.0.1
Port number: 5555
User name: pi
Password: ••••••••••••••••

☑ Show Login dialog on startup and when the last session is closed

0 B of 34.7 MB in 0 of 9
Not connected.