



Robot Vision II

With Python and OpenCV

Dr. Kourosh Pahlavan
Professor of the Practice, Robotics





Robot Vision I & II (MECH 296A I & II)

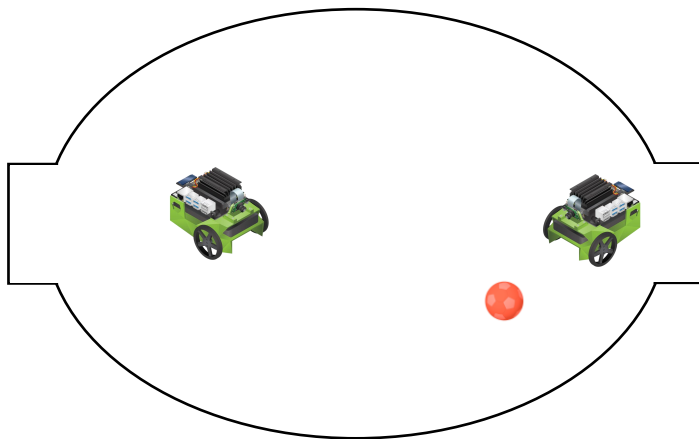
- Objectives:
 - Learn about fundamental techniques used in Computer Vision
 - Learn how to use OpenCV with Python and its tools
 - Learn the basics of how to use OpenCV to create robots that can “see”, “recognize” and locate objects.
 - Overall: learn the concepts and methods of using available tools
- Part I (MECH 296A I):
 - Learn basics of Computer Vision
 - Learn basics of OpenCV
 - Learn basics of Python
- Part II (MECH 296A II):
 - Bring your skills from part I to build robots that can recognize objects, locate, approach and relocate them
 - Build a “seeing” robot in teams of 2
 - Use your robots to compete in a Soccer match against other teams

Robot Vision I & II (MECH 246A-I & II)



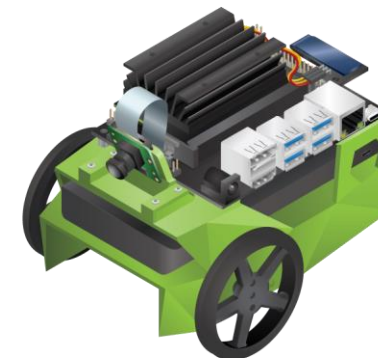
- Part I Ingredients:

- Python 3.6+ and IDE (Pycharm)
- OpenCV 4.5+
- Pre-trained object datasets, e.g. COCO
- Laptop with Webcam



- Part II Ingredients:

- Build a JetBot-like robot composed of:
 - NVIDIA Jetson Nano
 - Compatible camera
 - 2+ motors and drivers
 - Optional Arduino controller
 - Optional additional sensors



<https://jetbot.org>



Robot Vision II: Embedded Vision

- L1: Course Intro, Jetson Nano, Linux, Package Installations, group building
- L2: Basic Image Processing with Python on Jetson Nano
- L3: Camera Geometry, Motor Control and Interfaces + L2Q
- L4: Project Announcement and Workshop + L3Q
- L5: Project Workshop + GHWA1
- L6: Project Workshop + GHWA2
- L7: Project Workshop + GHWA3
- L8: Project Workshop + GHWA4
- L9: Hardware Disclosures and Formal Design Reviews
- L10: Robot Soccer Tournament

← Your robot must be fully operational at this point

Robot Vision II: Grading

Grading breakdown:

- **40%** Final Project (including individual reviews for grading)
- **30%** Group Homework Assignments (individual reviews)
- 10% Quizzes (including in class code reviews, no make-ups)
- 20% group/class engagement



Jetson Nano: Getting Started...



<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>

Included in the Box

Your Jetson Nano Developer Kit box includes:

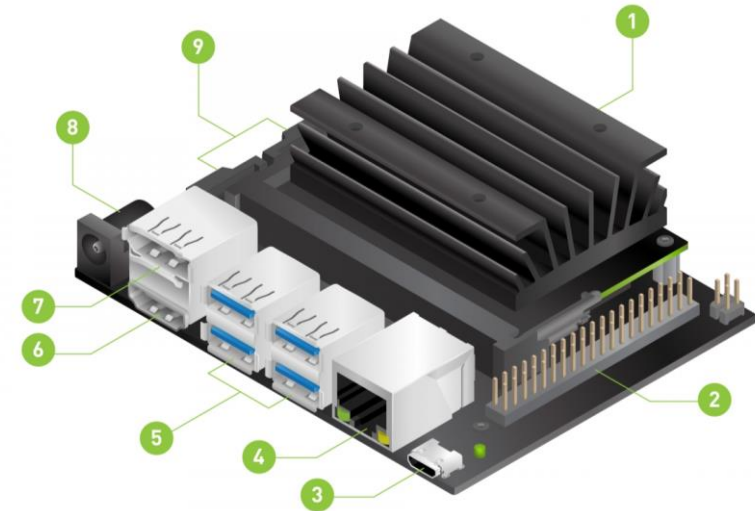
- NVIDIA Jetson module and reference carrier board
- Small paper card with quick start and support information
- Folded paper stand

Items not Included

You'll also need:

- microSD card (32GB UHS-1 minimum recommended)
- USB keyboard and mouse
- Computer display (HDMI or DP)
- Micro-USB power supply

+ wifi dongle

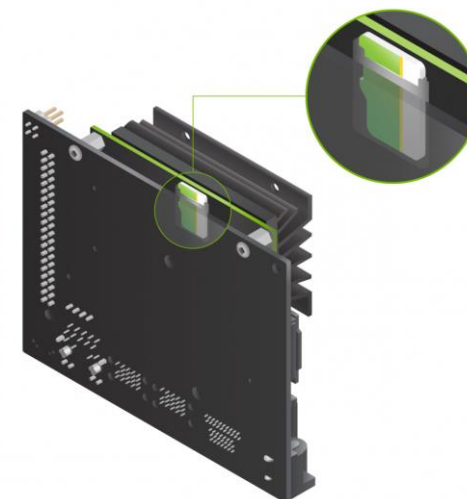


- | | |
|---|-------------------------------------|
| 1 microSD card slot for main storage | 5 USB 3.0 ports (x4) |
| 2 40-pin expansion header | 6 HDMI output port |
| 3 Micro-USB port for 5V power input, or for Device Mode | 7 DisplayPort connector |
| 4 Gigabit Ethernet port | 8 DC Barrel jack for 5V power input |
| | 9 MIPI CSI-2 camera connectors |



Jetson Nano: ...Getting Started

- Jetson boots from a SD card:
- Prepare a 32GB UHS-1 (minimum):
 - Download image
 - Format the card
 - Write image to SD card (OS-dependent)
- Insert the SD card
- Insert WiFi adapter
- Attach USB keyboard and mouse
- Attach USB adapter and fire up!



Robot Parts



- Place your orders in time:
 - Jetson Nano 2GB (4GB preferred)
 - IMX-219 camera (wide 160° or normal 77° lens)
 - https://www.amazon.com/SainSmart-IMX219-Camera-Module-Raspberry/dp/B07VFFRX4C/ref=sr_1_5?crid=808BE0ISAKMW&keywords=jetson+nano+camera&qid=1648661953&srefix=jetson+nano+camera%2Caps%2C171&sr=8-5
 - SD card (32GB UHS-1)
 - Mouse and Keyboard
 - WiFi adapter
 - Power supply (barrel or USB)
 - https://www.amazon.com/Waveshare-Power-Supply-Applicable-Jetson/dp/B07X8P1LFD/ref=sr_1_3?keywords=jetson%2Bnano%2Bpower%2Bsupply
 - https://www.amazon.com/Makeronics-100V-240V-Converter-Raspberry-Certificate/dp/B08DXZ1MSY/ref=sr_1_2?crid=1Q72EU8HEKJAC&keywords=jetson+nano+power+supply+micro+usb&qid=1648662471&srefix=jetson+nano+power+supply+micro+usb%2Caps%2C176&sr=8-2&qid=1648661828&sr=8-3&th=12

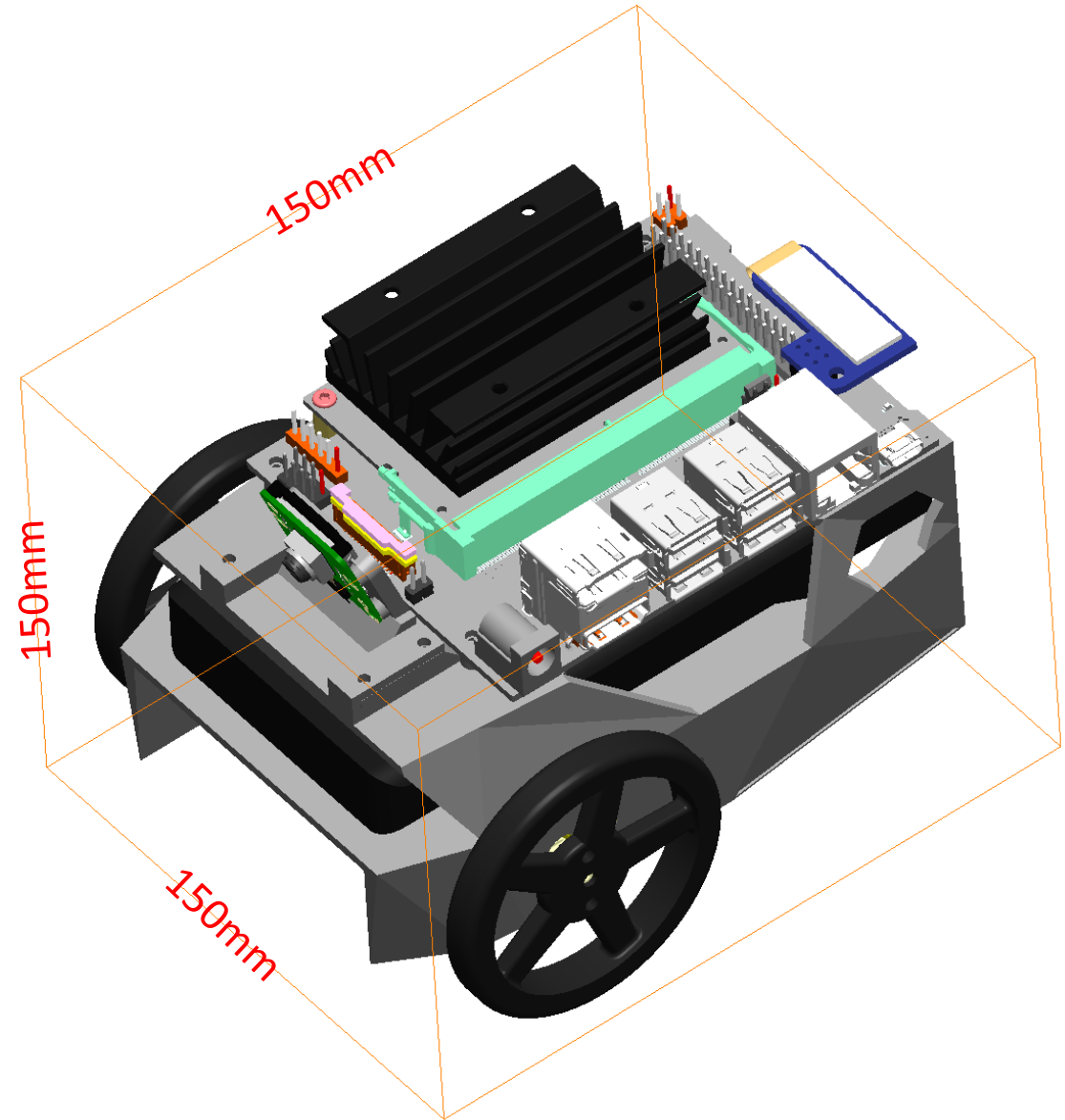
Motors and motor drivers:

- https://www.amazon.com/gp/product/B07T5SVCSX/ref=ewc_pr_img_1?smid=A30UAI9DCNKLVL&psc=1&pldnSite=1

- HDMI monitor
 - You can use any HDMI monitor incl. monitors in the classes and labs and your TV.
 - You can run your system headless
- Optional:
 - GPIO interface board
 - Arduino daughter board
 - Servos and other actuators
 - ...

Size Constraint

- Maximum dimensions:
 - 150mmx150mmx150mm
- No other physical constraint





Jetson Nano: WiFi adapters

- Jetson nano does not work with all dapters
 - Check compliance before purchase:
 - See: https://elinux.org/Jetson/Network_Adapters
 - Example: https://www.robotshop.com/en/jetson-nano-usb-wireless-network-card.html?gclid=EAlaIQobChMlvvuZwffn9gIVJA_nCh3MDw4KEAQYAiABEgJR9vD_BwE

- Fix intermittent disconnections (bug):

```
$ echo "blacklist rtl8192cu" | sudo tee -a /etc/modprobe.d/blacklist.conf
```

- Fix intermittent drops due to power saving:

```
$ sudo iw dev wlan0 set power_save off
```



Jetson Nano: Camera Module

- USB webcam
 - USB interface
- CSI-MIPI camera (IMX219 Raspberry Pi V2). Direct parallel interface -> faster

```
$ sudo apt-get update
```

```
$ sudo apt-get install v4l-utils
```

```
$ v4l2-ctl --list-formats-ext
```

```
ioctl: VIDIOC_ENUM_FMT
```

```
Index      : 0
```

```
Type       : Video Capture
```

```
Pixel Format: 'RG10'
```

```
Name       : 10-bit Bayer RGRG/GBGB
```

```
Size: Discrete 3280x2464 Interval: Discrete 0.048s (21.000 fps)
```

```
Size: Discrete 3280x1848 Interval: Discrete 0.036s (28.000 fps)
```

```
Size: Discrete 1920x1080 Interval: Discrete 0.033s (30.000 fps)
```

```
Size: Discrete 1280x720 Interval: Discrete 0.017s (60.000 fps)
```

```
Size: Discrete 1280x720 Interval: Discrete 0.017s (60.000 fps)
```

Camera flip parameters



`flip-method: video flip methods`

`flags: readable, writable, controllable`

`Enum "GstNvVideoFlipMethod" Default: 0, "none"`

- `(0): none` - Identity (no rotation)
- `(1): counterclockwise` - Rotate counter-clockwise 90 degrees
- `(2): rotate-180` - Rotate 180 degrees
- `(3): clockwise` - Rotate clockwise 90 degrees
- `(4): horizontal-flip` - Flip horizontally
- `(5): upper-right-diagonal` - Flip across upper right/lower left diagonal
- `(6): vertical-flip` - Flip vertically
- `(7): upper-left-diagonal` - Flip across upper left/low



Install and make all the prerequisites

- Source: <https://github.com/dusty-nv/jetson-inferencecv>
- Type the following lines in the shell sequentially:

```
$ sudo apt-get update
$ sudo apt-get install git cmake libpython3-dev python3-numpy
$ git clone --recursive https://github.com/dusty-nv/jetson-inference
$ cd jetson-inference
$ mkdir build
$ cd build
$ cmake ../
$ make -j$(nproc)
$ sudo make install
$ sudo ldconfig
```



Versions and Modules

- What OpenCV package is installed:

```
jetson:~$> dpkg -l | grep libopencv
```

- In Python 3 (note default is Python 2):

```
>>> import cv2
```

```
>>> print(cv2.getBuildInformation())
```

- Numpy installed?

```
>>> help ('modules')
```

- To install numpy:

<https://www.youtube.com/watch?v=dey9uTXuTo8>

```
jetson:~$> sudo apt-get install python3-numpy
```

- Check jet-pack version installed:

```
sudo apt-cache show nvidia-jetpack
```



Install Visual Studio Code

- Goto: <https://code.visualstudio.com/Download>
- Select ARM64 for Ubuntu (.deb)

```
$ sudo dpkg -i /home/nvidia/Downloads/code_.... (the version downloaded)
```

```
$ sudo apt install apt-transport-https
```

```
$ sudo apt update
```

```
$ sudo apt install code
```

```
$ code --version
```



15-Minute Linux Shell...

- Shell programs:

- Bourne shell (sh)
- C shell (csh)
- TC shell (tcsh)
- Korn shell (ksh)
- Bourne Again shell (bash)

- shell prompt:

```
user@jetson:~$
```

- Which shell you are running:

```
user@jetson:~$ echo $0
```

- Processes: `ps [s] [args]`

```
user@jetson:~$ ps -p <PID>
```

- Working directory: `pwd [s] [args]`

```
user@jetson:~$ pwd
```

- List directory: `ls [s] [arg]`

```
user@jetson:~$ ls /usr/local/bin
```

```
user@jetson:~$ man ls -al
```

```
user@jetson:~$ ls -al ~/mydir
```

- Change Directory: `cd [arg]`

```
user@jetson:~$ cd ./
```

```
user@jetson:~$ cd /
```

```
user@jetson:~$ cd ~
```

```
user@jetson:~$ cd ~/Desktop
```

```
user@jetson:~$ cd ..
```

```
user@jetson:~$ cd ../../
```

- User: `whoami`

```
user@jetson:~$ whoami
```




... 15-Minute Linux Shell

- Concatenate: `cat [s] [arg]`

```
user@jetson:~$ cat out.txt
```

```
user@jetson:~$ cat -b out.txt
```

- Copy: `cp [s] [args]`

```
user@jetson:~$ cp out.txt outcopy.txt
```

- Move and remove: `mv [s] [args]`

```
user@jetson:~$ mv out.txt ../          moves file!
```

```
user@jetson:~$ mv ./* ../tmp
```

- Make new directory: `mkdir [s] [args]`

```
user@jetson:~$ mkdir ./mydir/apps
```

```
user@jetson:~$ mkdir ./mydir/apps\ 1
```

```
user@jetson:~$ mkdir ./mydir/"apps 2"
```

- Remove: `rm, rmdir [s] [args]`

```
user@jetson:~$ rm out.txt removes file!
```

```
user@jetson:~$ rmdir mydir
```

```
user@jetson:~$ rm -d mydir
```

- Redirection to a file: `>`

```
user@jetson:~$ ls > out.txt          resets file!
```

```
user@jetson:~$ ls >> out.txt         appends file!
```



... 15-Minute Linux Shell

- "more" or "less": `less [s] [args]`

```
user@jetson:~$ more out.txt
```

```
user@jetson:~$ less out.txt
```

- Piping outputs to inputs: `|`

```
user@jetson:~$ ls ~/Desktop | less
```

```
user@jetson:~$ ls ~ | wc -l
```

```
user@jetson:~$ sort out.txt | uniq | wc -l
```

- Switch user: `su, sudo [args]`

```
user@jetson:~$ sudo apt-get install python3-numpy
```

- Arrow keys:
 - Recall previous commands
- Ctrl-C, Ctrl-z and Ctrl-d
 - Ctrl-c: Interrupt/kill the current foreground process running in the terminal. Not all processes will honor it.
 - Ctrl-z: (see also "bg") Suspend the current foreground process running in the shell. Use "fg <process_name>" to return the process to the foreground again.
 - Ctrl-d: Close the shell as does the exit command.

... 15-Minute Linux Shell

- File permissions:

```
$ ls -l filename.txt
-rw-r--r-- 12 krydda users 12.0K Apr  8 20:51 filename.txt
|[-][-][-]-  [-----] [---]
| | | | |      |      |
| | | | |      |      +-----> 7. Group
| | | | |      +-----> 6. Owner
| | | | +-----> 5. Links/subnodes
| | | +-----> 4. Others Permissions
| | +-----> 3. Group Permissions
| +-----> 2. Owner Permissions
+-----> 1. File Type
```

- Alphabetic:

```
$ chmod a-x filename
$ chmod og-rwx filename
```

- Nummeric:

- r (read) = 4
- w (write) = 2
- x (execute) = 1
- no permissions = 0

```
$ chmod 644 dirname
```

- Ownership:

```
$ chown class_296a file_name
$ chown another_user file1 dir1
```



ssh Connection...



- You can use ssh to securely and remotely connect to Jetson Nano:

```
Microsoft Windows [Version 10.0.19043.1586]
```

```
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\krydda>ssh 192.168.29.121
```

```
krydda@192.168.29.121's password:
```

```
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.9.253-tegra aarch64)
```

```
* Documentation:  https://help.ubuntu.com
```

```
* Management:    https://landscape.canonical.com
```

```
* Support:        https://ubuntu.com/advantage
```

```
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.
```

```
To restore this content, you can run the 'unminimize' command.
```

```
...
```

```
krydda@krydda-nano:~$ pwd
```

```
/home/krydda
```

```
krydda@krydda-nano:~$ exit
```

```
logout
```

```
Connection to 192.168.29.121 closed.
```



VNC Connection...

- You can use VNC to remotely connect to your Jetson Nano:
<https://developer.nvidia.com/embedded/learn/tutorials/vnc-setup>
- Type the following to set up your VNC server (already installed):

```
$ mkdir -p ~/.config/autostart
$ cp /usr/share/applications/vino-server.desktop ~/.config/autostart/.
$ cd /usr/lib/systemd/user/graphical-session.target.wants
$ sudo ln -s ../vino-server.service ../
$ gsettings set org.gnome.Vino prompt-enabled false
$ gsettings set org.gnome.Vino require-encryption false
$ gsettings set org.gnome.Vino authentication-methods "['vnc']"
$ gsettings set org.gnome.Vino vnc-password $(echo -n 'your_desired_password'|base64)
$ sudo reboot
```

- Jetson comes with "vino" as VNC server pre-installed. Also see:
<https://www.xmodulo.com/how-to-set-up-vnc-server-on-linux.html>



... VNC Connection

- Download your favorite VNC viewer from e.g.:
<https://www.realvnc.com/en/connect/download/viewer/>
- Find Jetson IP number to use in your VNC client:
`$ ifconfig`
- If you need the port number:
`$ sudo netstat -pl | grep vino`

Headless Connection

- Allows the user to employ any laptop as the host for Jetson
- See: README-usb-dev-mode.txt in L4T folder on Jetson desktop

