KENTECH
Korea Institute of Energy Technology

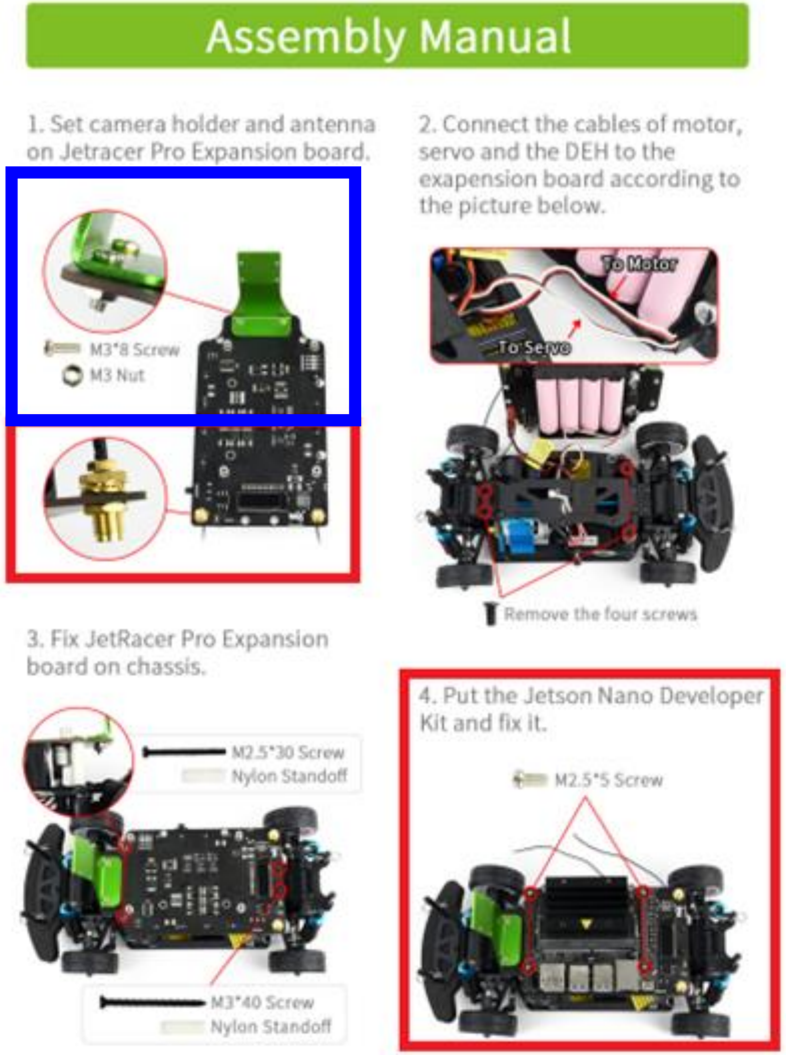
Visionary Course – Energy AI
Week 07

Apr. 19, 2022
Seokju Lee

KENTECH
Korea Institute of Energy Technology

Week 07b – Image Classification on Jetson Nano

Install Camera on Your Jetson Nano



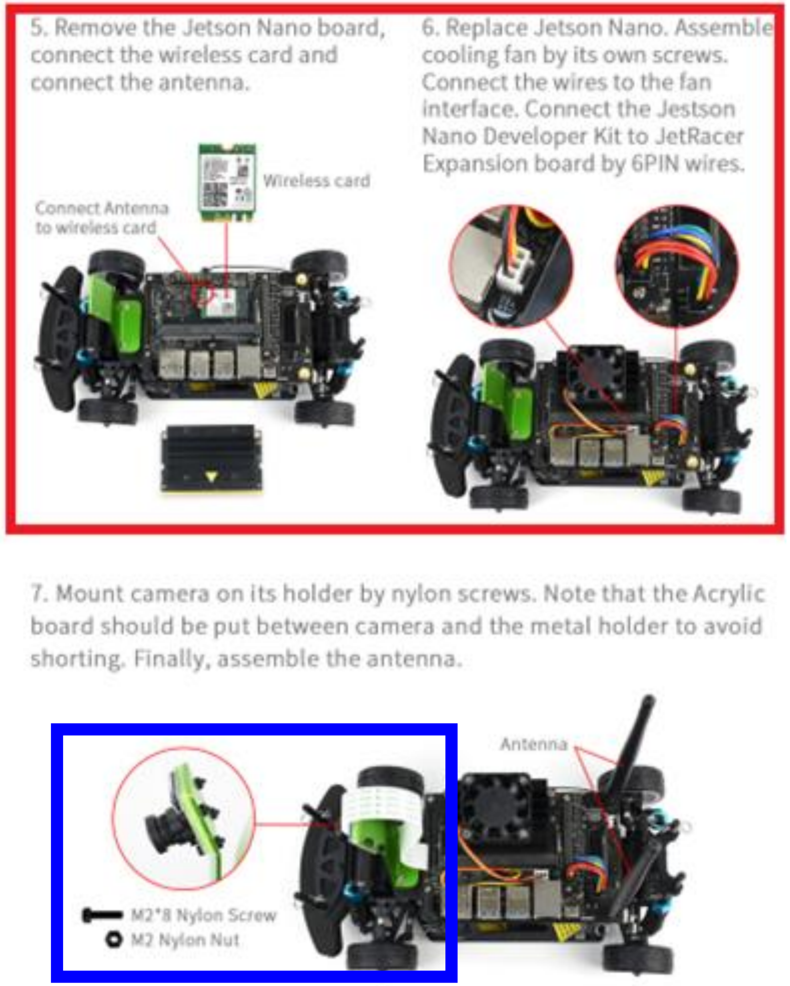
Assembly Manual

1. Set camera holder and antenna on Jetracer Pro Expansion board.

2. Connect the cables of motor, servo and the DEH to the exapension board according to the picture below.

3. Fix JetRacer Pro Expansion board on chassis.

4. Put the Jetson Nano Developer Kit and fix it.



5. Remove the Jetson Nano board, connect the wireless card and connect the antenna.

6. Replace Jetson Nano. Assemble cooling fan by its own screws. Connect the wires to the fan interface. Connect the Jestson Nano Developer Kit to JetRacer Expansion board by 6PIN wires.

7. Mount camera on its holder by nylon screws. Note that the Acrylic board should be put between camera and the metal holder to avoid shorting. Finally, assemble the antenna.

Follow the instructions in **blue boxes**!

Configurations

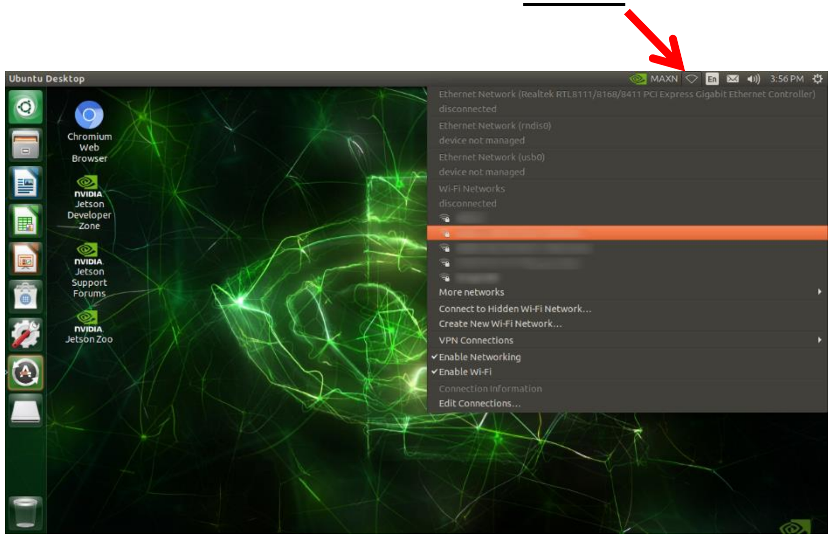
1. Connect the HDMI cable to the diplay.

2. Connect the keyboard and mouse to the Jetson Nano.

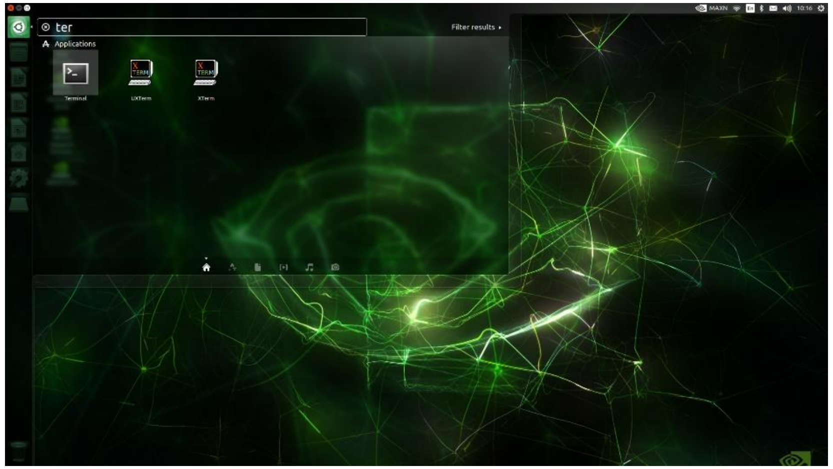
3. Please get used to the Linux interface (how to access file system, terminal, chrome, etc.).

   Basic Linux commands are listed in https://view.kentech.ac.kr/lecture/2022s/supp/ref01

4. Please connect to the WiFi.



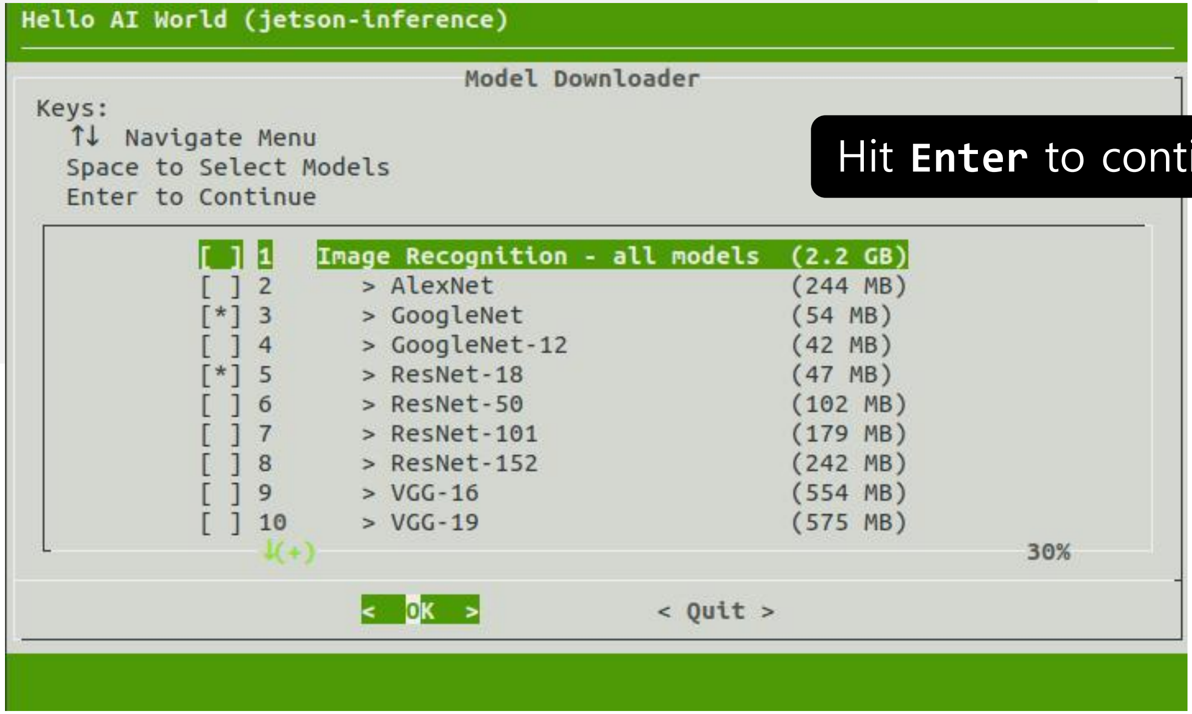
5. Open terminal.



Configurations: Hello-AI-World by NVIDIA

- Follow **Quick Reference** in https://github.com/dusty-nv/jetson-inference/blob/master/docs/building-repo-2.md

```
$ 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 -DENABLE_NVMM=OFF ../
$ make –j4
$ sudo make install
$ sudo ldconfig
```



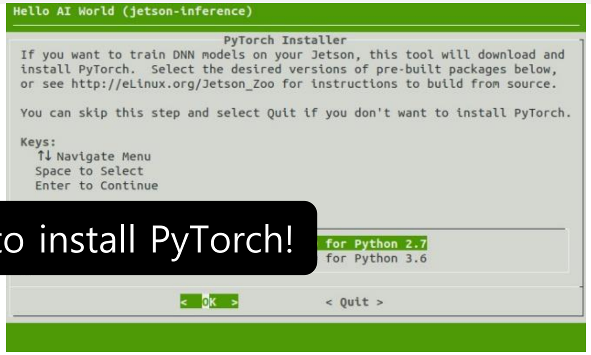
```
Hello AI World (jetson-inference)

                         Model Downloader
Keys:
  ↑↓ Navigate Menu
  Space to Select Models
  Enter to Continue

        [ ] 1   Image Recognition - all models   (2.2 GB)
        [ ] 2     > AlexNet                      (244 MB)
        [*] 3     > GoogleNet                    (54 MB)
        [ ] 4     > GoogleNet-12                 (42 MB)
        [*] 5     > ResNet-18                    (47 MB)
        [ ] 6     > ResNet-50                    (102 MB)
        [ ] 7     > ResNet-101                   (179 MB)
        [ ] 8     > ResNet-152                   (242 MB)
        [ ] 9     > VGG-16                       (554 MB)
        [ ] 10    > VGG-19                       (575 MB)
            ↓(+)                                          30%

              <  OK  >              < Quit >
```

Hit **Enter** to continue!



```
Hello AI World (jetson-inference)

                    PyTorch Installer
If you want to train DNN models on your Jetson, this tool will download and
install PyTorch.  Select the desired versions of pre-built packages below,
or see http://eLinux.org/Jetson_Zoo for instructions to build from source.

You can skip this step and select Quit if you don't want to install PyTorch.

Keys:
  ↑↓ Navigate Menu
  Space to Select
  Enter to Continue

                                  for Python 2.7
                                  for Python 3.6

            <  OK  >            < Quit >
```

You don't need to install PyTorch!

Experiments

### Before Starting ###

*Your basic workspace is here: “cd ~/jetson-inference/build/aarch64/bin” Every code is pre-built in this path.

### Video Streaming ###

Q1.1. Run “python video-viewer.py csi://0” What is the output?

Q1.2. Run “python video-viewer.py --flip-method=rotate-180 csi://0” Discuss the differences.

Q1.3. Run “python video-viewer.py --flip-method=rotate-180 --input-width=640 --input-height=480 csi://0” Discuss the differences.

Q1.4. Run “python video-viewer.py --flip-method=rotate-180 --input-width=640 --input-height=480 --framerate=10 csi://0” Discuss the differences.

### Live Demo for Image Classification ###

Q2.1. Run “python imagenet.py --flip-method=rotate-180” What is the output of the pop-up display? Let’s check the terminal output. Please take a screenshot and paste it here. You can see some output values. What does each output (network name, class ID, floating-point number next to it, class name, each processing time, etc.) mean?

Experiments

Q2.2. Go to the linked page (https://deeplearning.cms.waikato.ac.nz/user-guide/class-maps/IMAGENET/) and check that the class ID is matched to the class name. How many classes can the model distinguish in total? Please prepare your **own object (🕶,👜,🌂,👢,⚾)** corresponding to one of the above classes for further experiments (classification, detection, etc.).

Q2.3. Run "cd ~/jetson-inference/build; ./download-models.sh;" to download different CNN models (e.g., AlexNet, ResNet-50, etc.). Run "python imagenet.py --flip-method=rotate-180 --network=resnet-50". Please check the qualitative performance of each model.

### Classify Your Own Objects or Images ###

Q3.1. Place the object in front of the camera and run the code (imagenet.py). Please take a screenshot of the result.

Q3.2. Position the object closer or further away from the camera. Please Analyze how confidence changes.

Q3.3. Download random images from Google and classify them. Please refer "python my-recognition.py images/banana_0.jpg --network=googlenet" and the below code:

```
import PIL
img = PIL.Image.open('jellyfish.jpg').resize((224,224))
img = np.array(img)
img_cuda = jetson.utils.cudaFromNumpy(img)      # CUDA image
class_id, confidence = net.Classify(img_cuda)   # Inference
class_desc = net.GetClassDesc(class_id)         # Predicted class
print(class_desc, confidence)
```

Q3.4. Please try other CNN models and repeat Q3.

Experiments

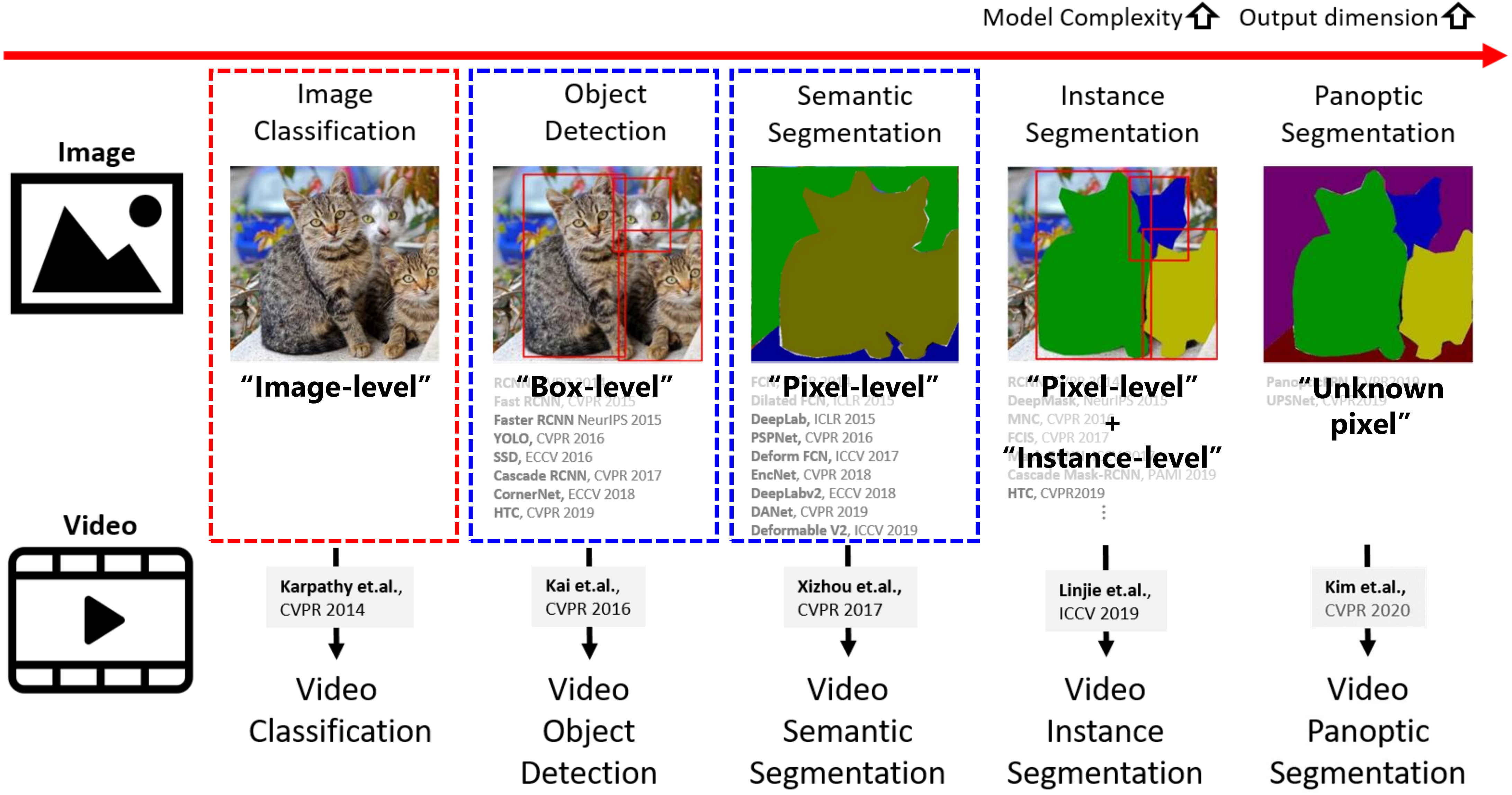
### Some Useful Tips while Debugging ###

*Sometimes, the python process does not respond. In this case, please terminate the process with `ctrl+c`. If it still does not respond at all, forcibly stop the process with `ctrl+z`, and check the running process name with the `ps -a` command, and then type `sudo pkill -9 [name-of-process]` command to kill the process. If you don't shut it down, it will remain as a 🧟zombie🧟 and keep occupying the processor (CPU or GPU) in the background.

*Sometimes, the best solution for resolving an issue is just rebooting the system.

Computer Vision Tasks

*Slide by Kim, et al., “Video Panoptic Segmentation” (CVPR 2020)



Model Complexity ⇧ Output dimension ⇧

Image

Image Classification

“Image-level”

Object Detection

“Box-level”

RCNN, CVPR 2014
Fast RCNN, CVPR 2015
Faster RCNN NeurIPS 2015
YOLO, CVPR 2016
SSD, ECCV 2016
Cascade RCNN, CVPR 2017
CornerNet, ECCV 2018
HTC, CVPR 2019

Semantic Segmentation

“Pixel-level”

FCN, CVPR 2014
Dilated FCN, ICLR 2015
DeepLab, ICLR 2015
PSPNet, CVPR 2016
Deform FCN, ICCV 2017
EncNet, CVPR 2018
DeepLabv2, ECCV 2018
DANet, CVPR 2019
Deformable V2, ICCV 2019

Instance Segmentation

“Pixel-level”
+
“Instance-level”

RCNN, CVPR 2014
DeepMask, NeurIPS 2015
MNC, CVPR 2016
FCIS, CVPR 2017
Mask-RCNN, ICCV 2017
Cascade Mask-RCNN, PAMI 2019
HTC, CVPR2019
⋮

Panoptic Segmentation

“Unknown pixel”

Panoptic FPN, CVPR2019
UPSNet, CVPR2019

Video

Karpathy et.al., CVPR 2014 → Video Classification

Kai et.al., CVPR 2016 → Video Object Detection

Xizhou et.al., CVPR 2017 → Video Semantic Segmentation

Linjie et.al., ICCV 2019 → Video Instance Segmentation

Kim et.al., CVPR 2020 → Video Panoptic Segmentation