Intro to AI Platform

Lab4 : Jetson

Hyungmin Kim, Sihwa Lee

(kong4274@hanyang.ac.kr, macto94@hanyang.ac.kr)

Hanyang University

2022 Spring

Basic Settings on Jetson Nano

- install python3-pip

```
$sudo apt install python3-pip
```

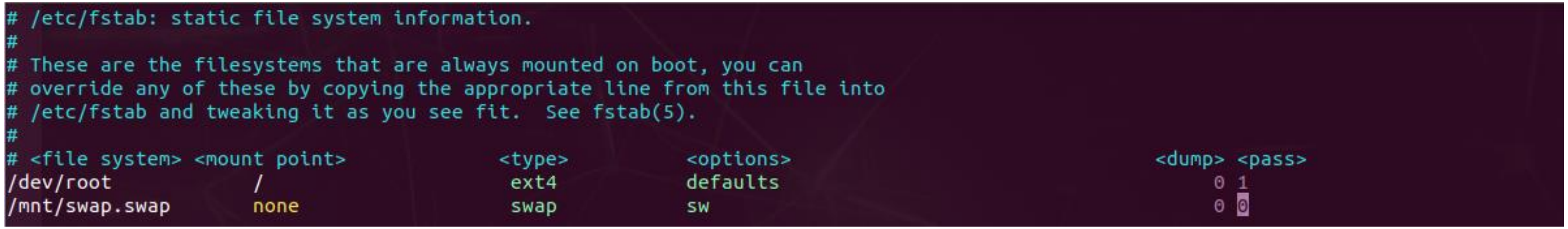
- Memory Swap

  : Since Memory(4GB) on the Jetson Nano is rather limited, create and mount a swap file on the System.

```
$sudo fallocate -l 8G /mnt
$sudo mkswap /mnt/8GB.swap
$sudo swapon /mnt/8GB.swap
```

- add following line into '/etc/fstab/ and reboot the system. It Make sure the swap space gets mounted automatically after reboot

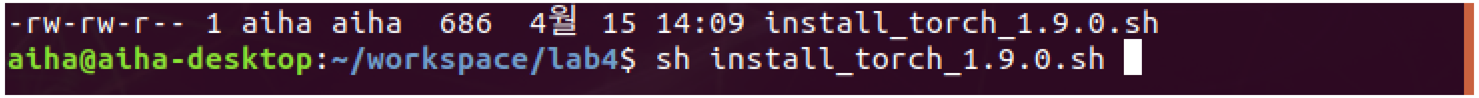
```
/mnt/8GB.swap none swap sw 0 0
```



```
# /etc/fstab: static file system information.
#
# These are the filesystems that are always mounted on boot, you can
# override any of these by copying the appropriate line from this file into
# /etc/fstab and tweaking it as you see fit.  See fstab(5).
#
# <file system> <mount point>             <type>          <options>                               <dump> <pass>
/dev/root            /                     ext4           defaults                                     0 1
/mnt/swap.swap       none                  swap           sw                                           0 0
```

Install PyTorch on Jetson Nano

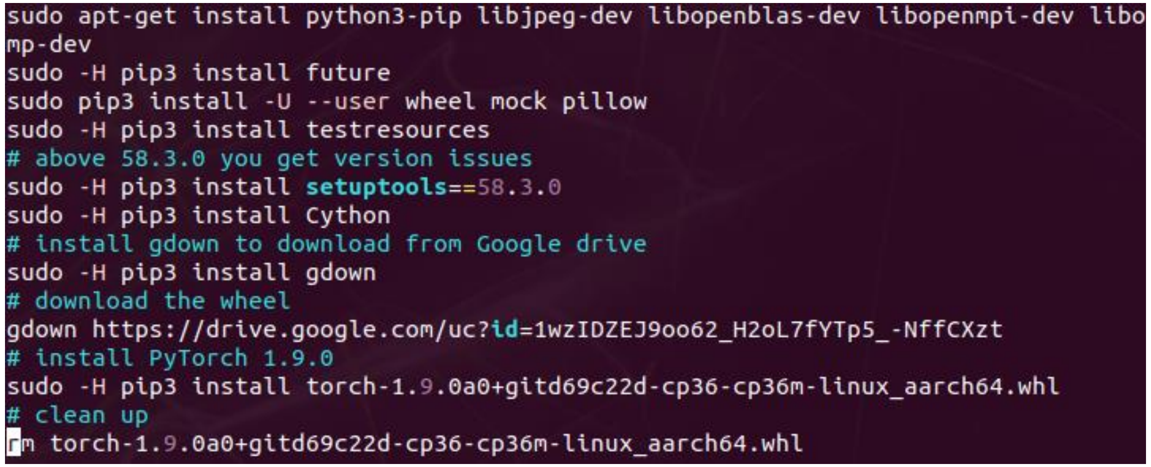
1. Install PyTorch 1.9.0



```
-rw-rw-r-- 1 aiha aiha  686  4월 15 14:09 install_torch_1.9.0.sh
aiha@aiha-desktop:~/workspace/lab4$ sh install_torch_1.9.0.sh
```

```
$sh install_torch_1.9.0.sh
```

: execute distributed shell script file on your device.



```
sudo apt-get install python3-pip libjpeg-dev libopenblas-dev libopenmpi-dev libo
mp-dev
sudo -H pip3 install future
sudo pip3 install -U --user wheel mock pillow
sudo -H pip3 install testresources
# above 58.3.0 you get version issues
sudo -H pip3 install setuptools==58.3.0
sudo -H pip3 install Cython
# install gdown to download from Google drive
sudo -H pip3 install gdown
# download the wheel
gdown https://drive.google.com/uc?id=1wzIDZEJ9oo62_H2oL7fYTp5_-NffCXzt
# install PyTorch 1.9.0
sudo -H pip3 install torch-1.9.0a0+gitd69c22d-cp36-cp36m-linux_aarch64.whl
# clean up
rm torch-1.9.0a0+gitd69c22d-cp36-cp36m-linux_aarch64.whl
```

2. Install torchvision 0.10.0 (compatible with PyTorch 0.10.0)

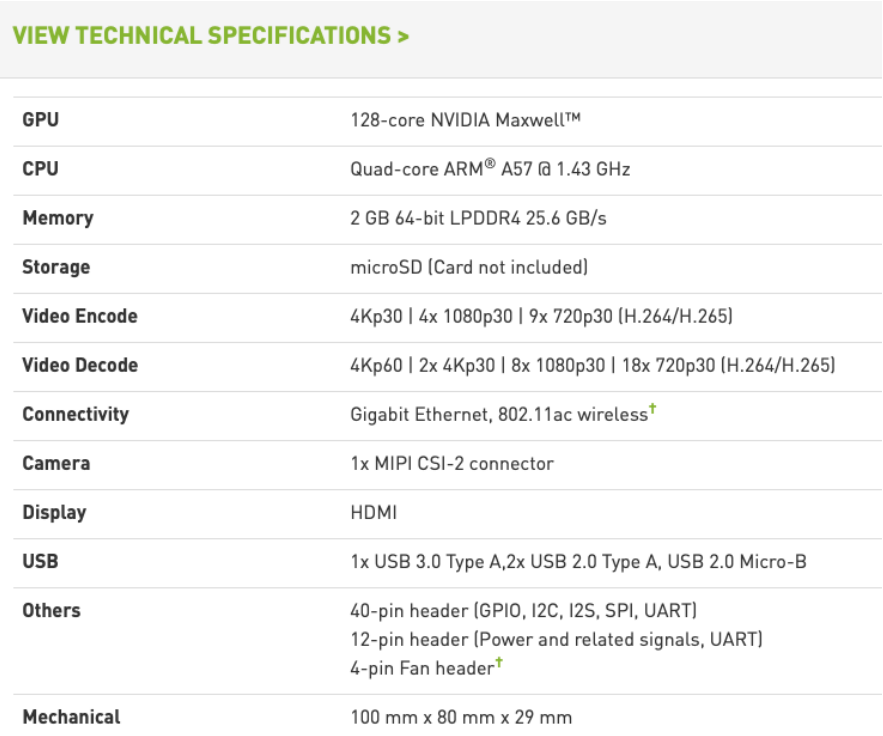
```
$sh install_torchvision_0.10.0.sh
```

NVIDIA Jetson Nano

: a series of embedded computing boards from Nvidia

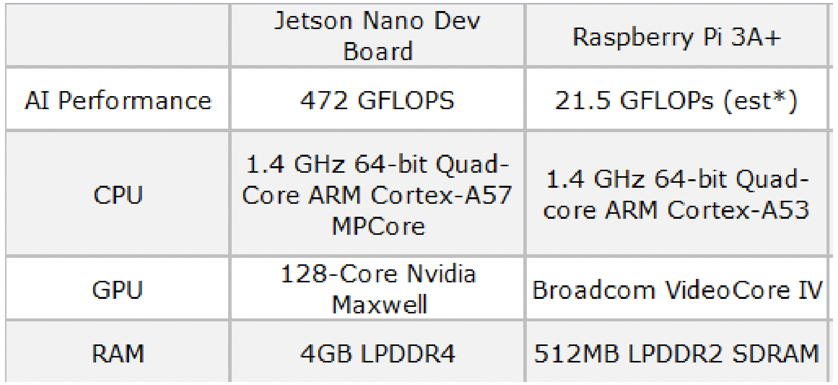
: a low-power system and is designed for “**accelerating**” deep learning applications

: is used for hands-on AI learning and making projects



VIEW TECHNICAL SPECIFICATIONS >

GPU	128-core NVIDIA Maxwell™
CPU	Quad-core ARM® A57 @ 1.43 GHz
Memory	2 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (Card not included)
Video Encode	4Kp30 | 4x 1080p30 | 9x 720p30 (H.264/H.265)
Video Decode	4Kp60 | 2x 4Kp30 | 8x 1080p30 | 18x 720p30 (H.264/H.265)
Connectivity	Gigabit Ethernet, 802.11ac wireless†
Camera	1x MIPI CSI-2 connector
Display	HDMI
USB	1x USB 3.0 Type A,2x USB 2.0 Type A, USB 2.0 Micro-B
Others	40-pin header (GPIO, I2C, I2S, SPI, UART) 12-pin header (Power and related signals, UART) 4-pin Fan header†
Mechanical	100 mm x 80 mm x 29 mm



	Jetson Nano Dev Board	Raspberry Pi 3A+
AI Performance	472 GFLOPS	21.5 GFLOPs (est*)
CPU	1.4 GHz 64-bit Quad-Core ARM Cortex-A57 MPCore	1.4 GHz 64-bit Quad-core ARM Cortex-A53
GPU	128-Core Nvidia Maxwell	Broadcom VideoCore IV
RAM	4GB LPDDR4	512MB LPDDR2 SDRAM

Jetson Nano – Jetpack 4.6

Jetpack : the solution for building AI applications. it includes :

- OS image : a reference file system derived from Ubuntu

- Libraries : TensorRT , cuDNN, CUDA, Multimedia API, OpenCV etc.

These libraries are already installed, so you don’t need to install basic libraries.

TensorRT : SDK(Software Development Kit) for deep learning inference

cuDNN : GPU-accelerated library for DNN

OpenCV : Open source library for computer vision, image processing

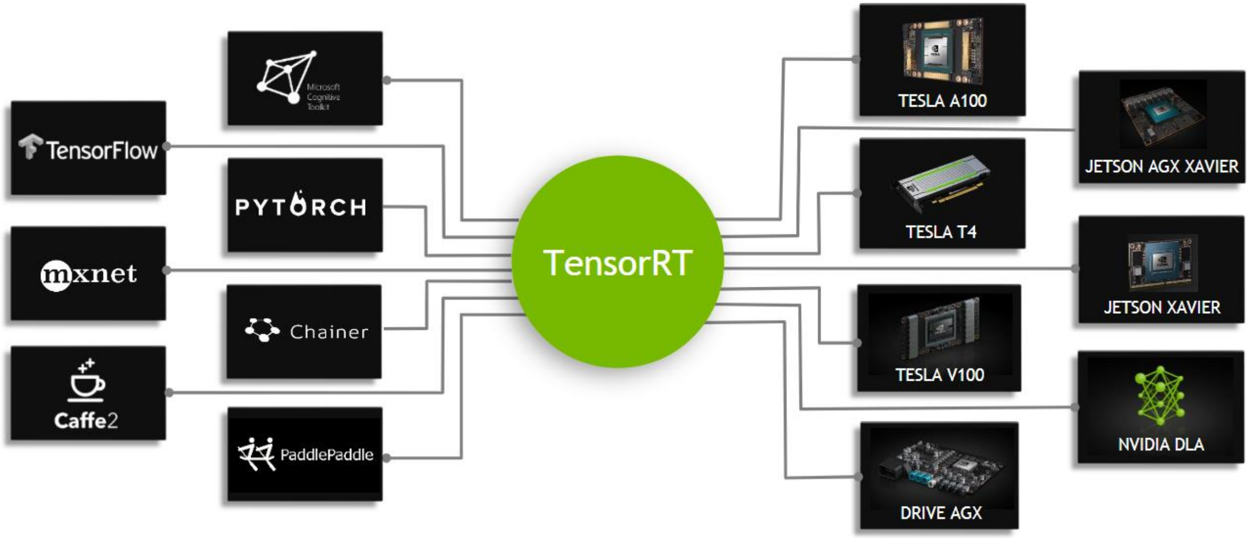
Jetpack 4.6 – TensorRT

TensorRT

: C++, Python library that facilitates high-performance on NVIDIA’s GPUs

: complementary with frameworks TensorFlow, Pytorch, Caffe, etc.

: quick and efficient for the purpose of generating result such as scoring, detecting, regression, or inference



Jetpack 4.6 – CUDA, cuDNN

CUDA

: NVIDIA GPUs’ API

: for GPU accelerated applications across multiple domains

: includes a compiler(nvcc) for NVIDIA GPUs, math libraries, and tools for optimizing

cuDNN (NVIDIA CUDA Deep Neural Network library)

: GPU-accelerated library

: enables us to focus on training NN and developing software applications rather than spending time on low-level GPU performance tuning

Jetpack 4.6 – OpenCV

OpenCV (Open source Computer Vision library)

: open source library for computer vision, image processing (e.g., camera control) etc.

: Functionality

- Image/video I/o, processing, display
- Object/feature detection
- etc

: easy to use, lots of examples are on the web

OpenCV - Tutorial

OpenCV's Official Documentation
(https://docs.opencv.org/4.x/)

OpenCV (Open Source Computer Vision Library) is an open-source library that includes several hundreds of computer vision algorithms.

In the Lab4 folder, you can test the CV algorithms

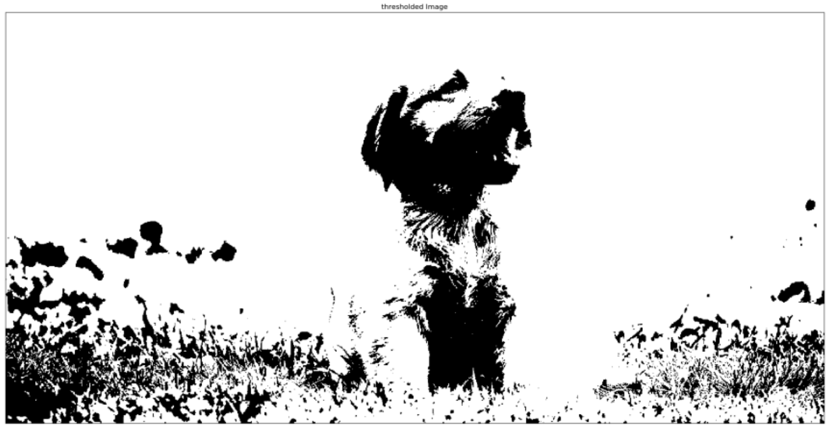
```
python3 thresholding.py
```

```
python3 canny_edge_detection.py
```

```
python3 background_sub.py
```

OpenCV - Tutorial

1. Image Thresholding



```
cv2.threshold(image, threshold_value, max_value, flag)
```

: If pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value.

flag:

cv2.THRESH_BINARY :

$$dst(x,y) = \begin{cases} \texttt{maxValue} & \text{if } src(x,y) > T(x,y) \\ 0 & \text{otherwise} \end{cases}$$

cv2.THRESH_BINARY_INV :

$$dst(x,y) = \begin{cases} 0 & \text{if } src(x,y) > T(x,y) \\ \texttt{maxValue} & \text{otherwise} \end{cases}$$

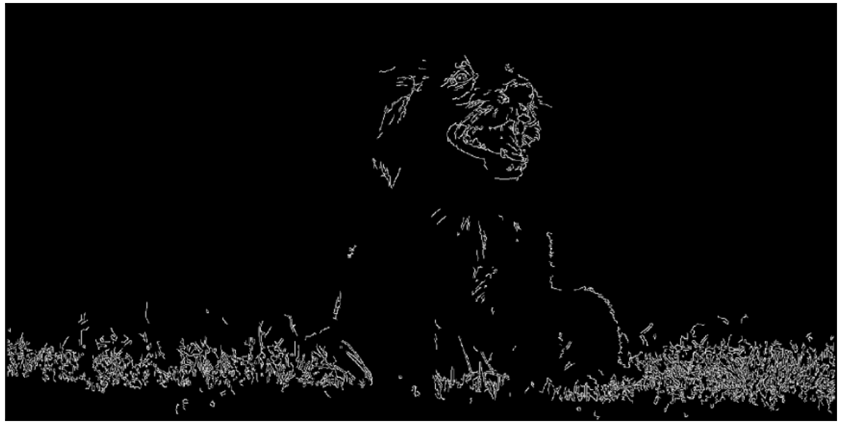
cv2.THRESH_TRUNC :

$$\texttt{dst}(x,y) = \begin{cases} \texttt{threshold} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$$

…

OpenCV - Tutorial

2. Canny Edge Detection



1.Apply Gaussian filter to smooth the image in order to remove the noise

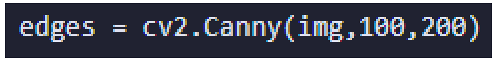
2.Find the intensity gradients of the image

3.Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection

4.Apply double threshold to determine potential edges

5.Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

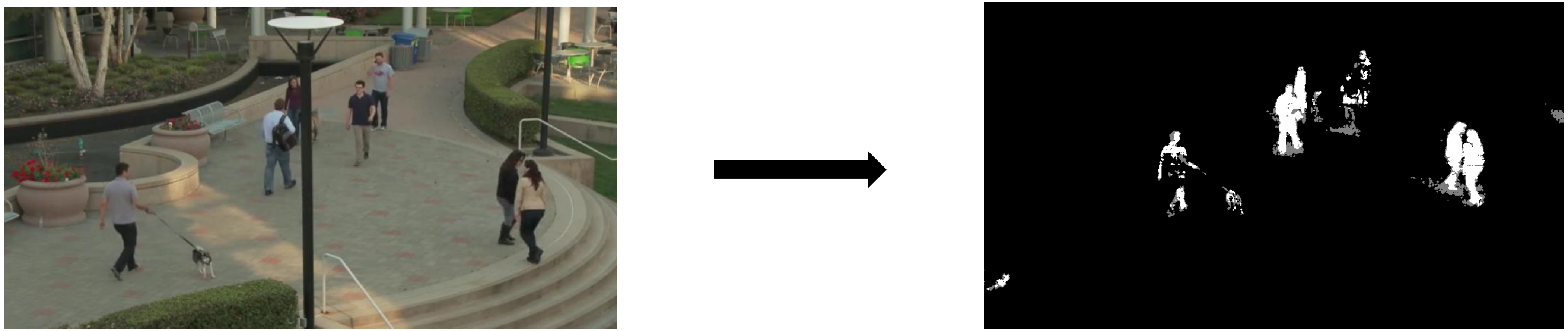
You can implement complicated algorithm in a single instruction!



```
edges = cv2.Canny(img,100,200)
```

OpenCV - Tutorial

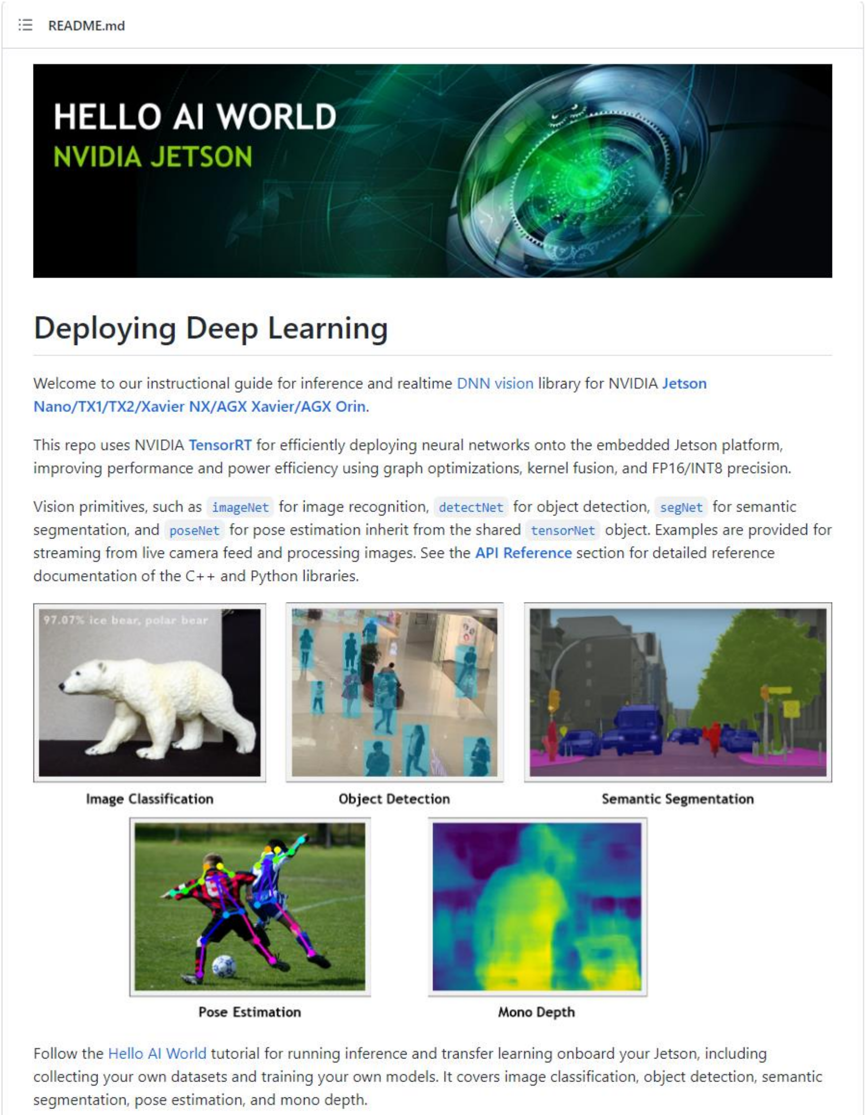
3. Background Subtraction



- **Background Subtraction(BS) is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras**
- **In this example, We use the cv2. BackgroundSubtractorMOG2**

BackgroundSubtractorMOG2 – *It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It provides better adaptability to varying scenes due illumination changes etc.*

Hello AI World- jetson-inference



Deploying Deep Learning

Welcome to our instructional guide for inference and realtime DNN vision library for NVIDIA Jetson Nano/TX1/TX2/Xavier NX/AGX Xavier/AGX Orin.

This repo uses NVIDIA TensorRT for efficiently deploying neural networks onto the embedded Jetson platform, improving performance and power efficiency using graph optimizations, kernel fusion, and FP16/INT8 precision.

Vision primitives, such as `imageNet` for image recognition, `detectNet` for object detection, `segNet` for semantic segmentation, and `poseNet` for pose estimation inherit from the shared `tensorNet` object. Examples are provided for streaming from live camera feed and processing images. See the API Reference section for detailed reference documentation of the C++ and Python libraries.

Image Classification

Object Detection

Semantic Segmentation

Pose Estimation

Mono Depth

Follow the Hello AI World tutorial for running inference and transfer learning onboard your Jetson, including collecting your own datasets and training your own models. It covers image classification, object detection, semantic segmentation, pose estimation, and mono depth.

- Codebase that contains a lot of AI Tasks
  (https://github.com/dusty-nv/jetson-inference)
  - Image Classification
  - Object Detection
  - Semantic Segmentation
  - Pose Estimation
  - Mono Depth

- For your final Project, this codebase will be very helpful for you project

Hello AI World- Getting Started

1. With your Jetson nano,

```
$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
$sudo cmake ..
$sudo make -j4 (4 is optimal)
$sudo make install
$sudo ldconfig
```

2. Using the ImageNet Program on Jetson

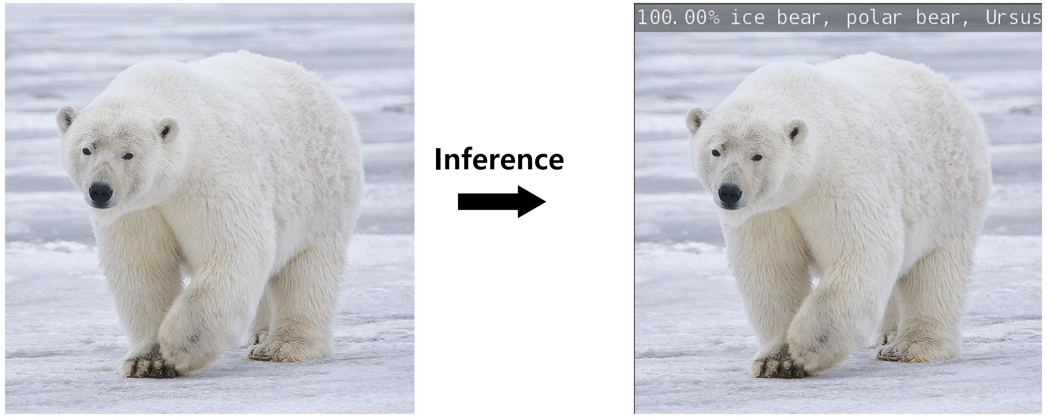
```
#python
$./imagenet.py (input_video_path) (output_video_path)
$./imagenet.py images/fruit_0.jpg images/test/output_0.jpg
```

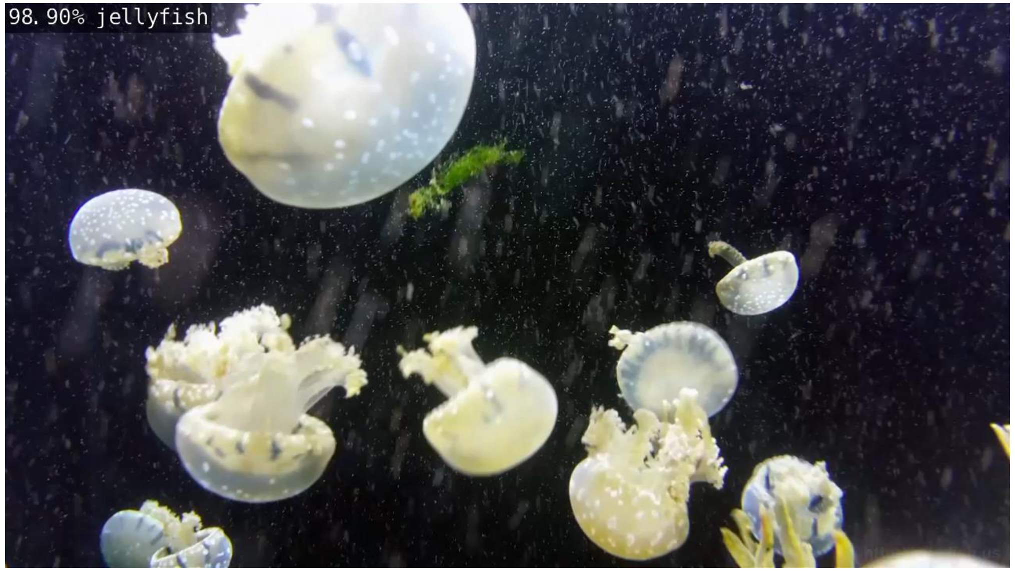
Hello AI World- Classification

Pre-Trained Model: GoogleNet

- The First time you run each model, It will take a few minutes to build the network
- You can classify images or video whatever you want

```
#Video Processing
ex)
$wget https://nvidia.box.com/shared/static/tlswont1jnyu3ix2tbf7utaekpzcx4rc.mkv%20-O%20jellyfish.mkv
#ResNet-18 based Image Classification
$./imagenet.py –network=resnet-18 jellyfish.mkv images/test/jellyfish_resnet18.mkv
```





Lab 4 Assignment

- Try two more OpenCV tutorial (include Thresholding, edge detection, background subtraction)

  (reference: https://docs.opencv.org/3.4/d9/df8/tutorial_root.html)

- So, you should have to submit the Report about 5 tutorials

- If you have any difficult to understand codes we handled in Lab class, please include it in report

  - Submit a report by Apr. 25th (via LMS)

    - Less than 3 pages, free format