KENTECH
Korea Institute of Energy Technology

Visionary Course – Energy AI
Week 07

Apr. 15, 2022

Seokju Lee

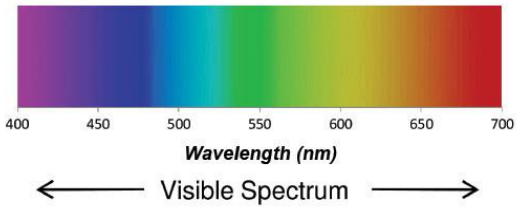
KENTECH
Korea Institute of Energy Technology

Week 07a – Visual Perception

Human Visual System

"About **half** of neocortex in humans is devoted to **vision**." [1]

Low-level visual signal



(2) **Geometric perception**

Where is it?

Two independent visual pathways:

"**What**-path & **Where**-path"

High-level decision

(1) **Semantic perception**

What is it?

[1] Barton, Robert A. "Visual specialization and brain evolution in primates." *Proceedings of the Royal Society of London* (1998).
[2] M. A. Goodale, et al., "Separate visual pathways for perception and action." *Trends in Neurosciences* (1992).

Visual Perception: Semantics & Geometry

"Semantic" perception

: *Meaning* of an element, *syntax*, *context* of scene, or *relationship* between objects.

"Geometric" perception

: *Distance*, *shape*, *structure*, *size*, *scale* of an element, *3D space* where we live, *relative position* between objects.

Semantic computer vision tasks

- Image classification
- Object detection
- Semantic segmentation
- …

Geometric computer vision tasks

- Depth estimation
- Pose estimation
- 3D reconstruction
- …

Video understanding

ex) Video classification

+ "Temporal"

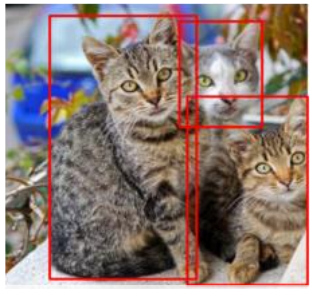
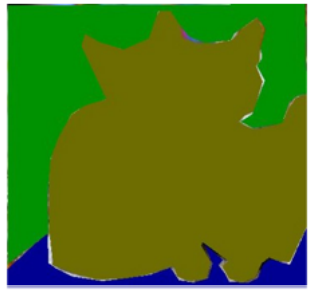
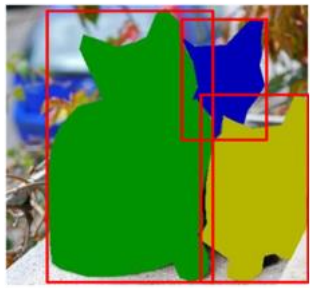
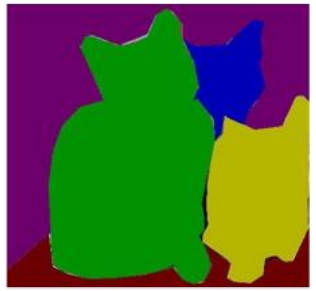
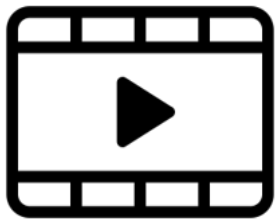
Motion understanding

ex) 3D motion estimation

Computer Vision Tasks

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

Model Complexity ⇧ Output dimension ⇧

	Image Classification	Object Detection	Semantic Segmentation	Instance Segmentation	Panoptic Segmentation
Image					
					
	"Image-level"	"Box-level"	"Pixel-level"	"Pixel-level" + "Instance-level"	"Unknown pixel"
Video					
	Karpathy et.al., CVPR 2014	Kai et.al., CVPR 2016	Xizhou et.al., CVPR 2017	Linjie et.al., ICCV 2019	Kim et.al., CVPR 2020
	Video Classification	Video Object Detection	Video Semantic Segmentation	Video Instance Segmentation	Video Panoptic Segmentation

KENTECH
Korea Institute of Energy Technology

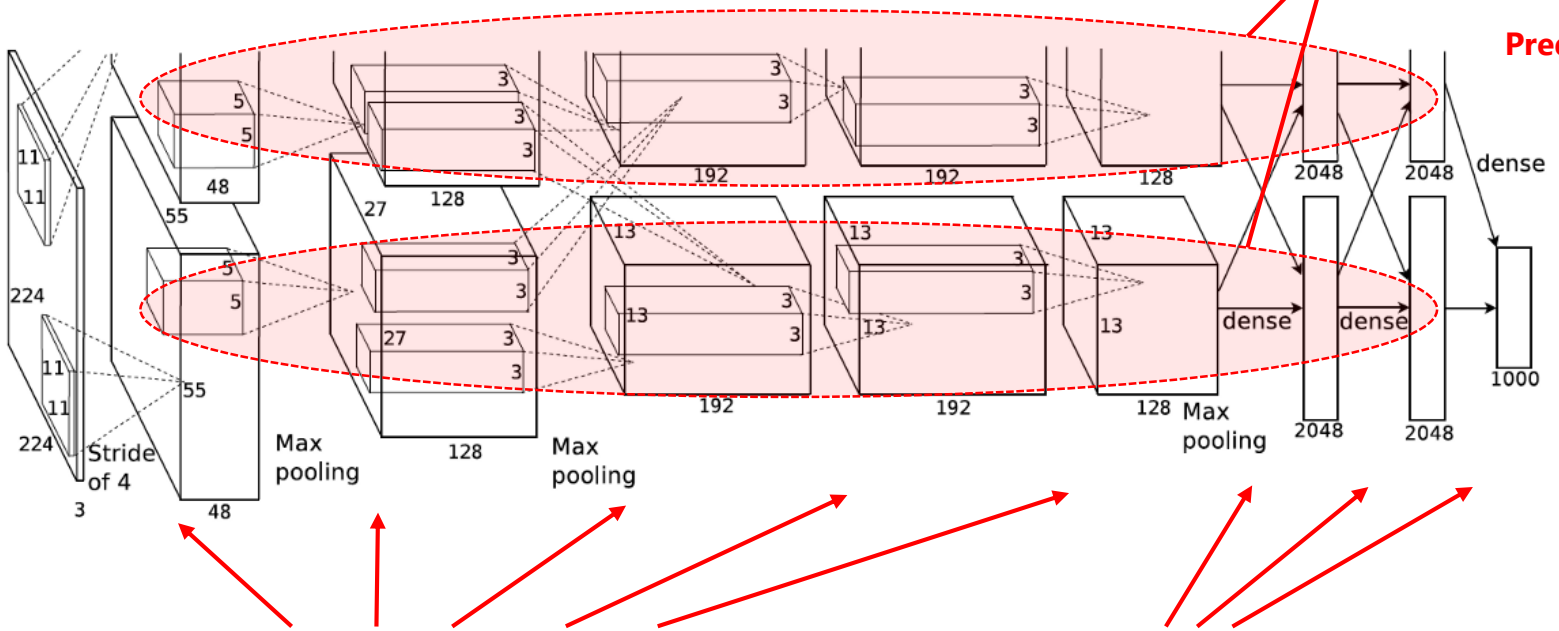
Week 07a – Image Classification

Image Classification

The most fundamental task using deep learning!

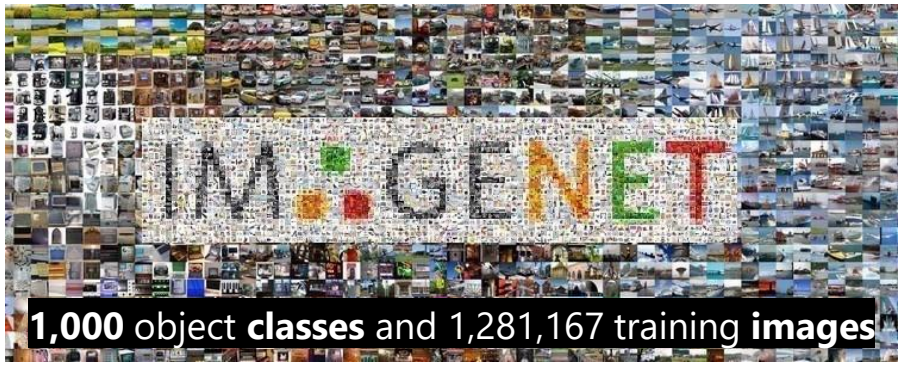
→ What's going on inside deep neural networks?

→ AlexNet [1] (# cites: **106,100** in Mar. 2022)



[1] ImageNet Classification with Deep Convolutional Neural Networks; NIPS 2012

AlexNet: Breakthrough in 2012

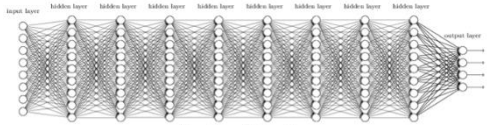


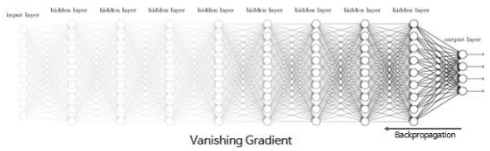
Troublesome of previous neural networks

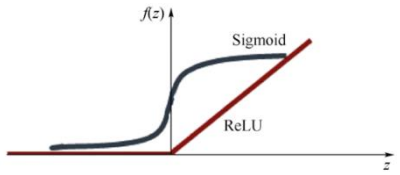
- Local minimum or slow learning
- Overfitting
- Small data
- Time complexity
- Vanishing gradients

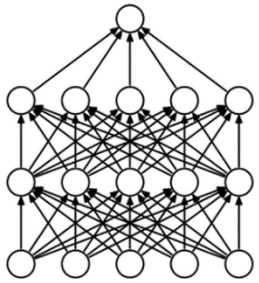
AlexNet

- Big data: ImageNet Challenge
- GPUs
- ReLU
- Dropout
- Deeper network

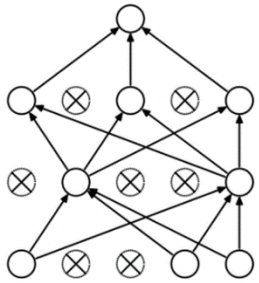




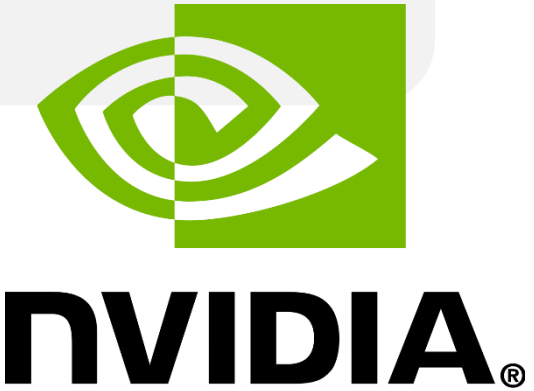




(a) Standard Neural Net



(b) After applying dropout.



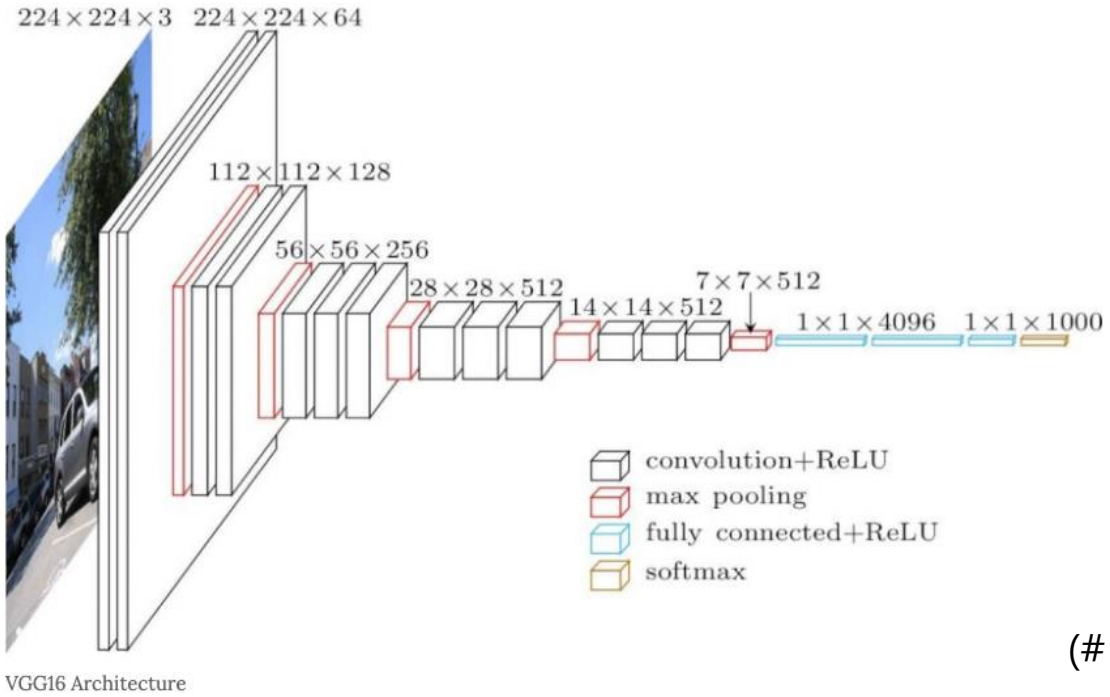
[1] ImageNet Classification with Deep Convolutional Neural Networks; NIPS 2012

VGG: ImageNet Challenge (2014) 2nd Place

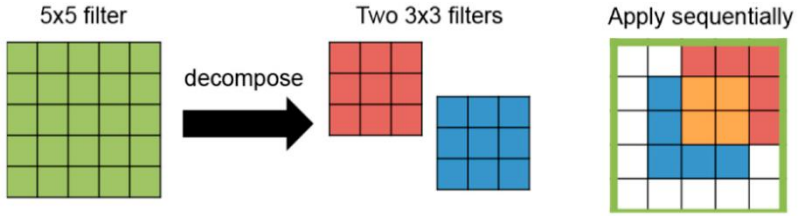
Small filters + Deeper networks + Beautifully uniform design

→ Why use smaller filters?

Number of parameters ↓ (efficiency ↑) + Deeper layer (nonlinearity ↑)



VGG16 Architecture



5x5 = **25** > 3x3x2 = **18**

(# cites: **76,400** in Mar. 2022)

[1] Very Deep Convolutional Networks for Large-Scale Image Recognition; 2015; ICLR 2015

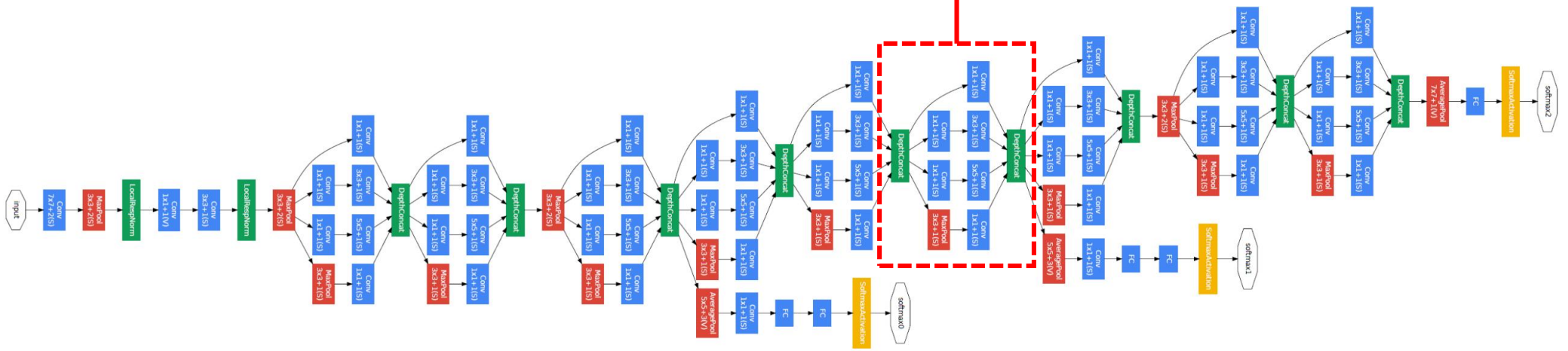
GoogleNet: ImageNet Challenge (2014) Winner

Deeper networks with a computational efficiency

→ **Inception** module: Local network topology (network within a network)

→ 5M params. (×12 less than AlexNet)

→ Fully convolutional networks



(# cites: **38,700** in Mar. 2022)

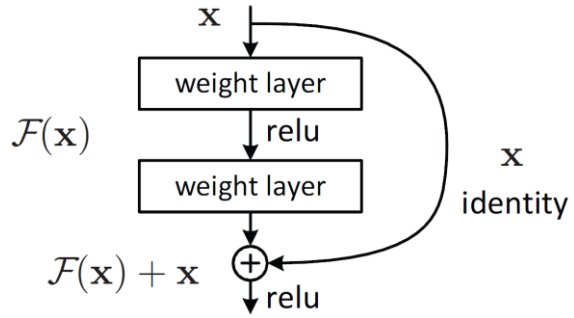
[1] Going Deeper with Convolutions, CVPR 2015

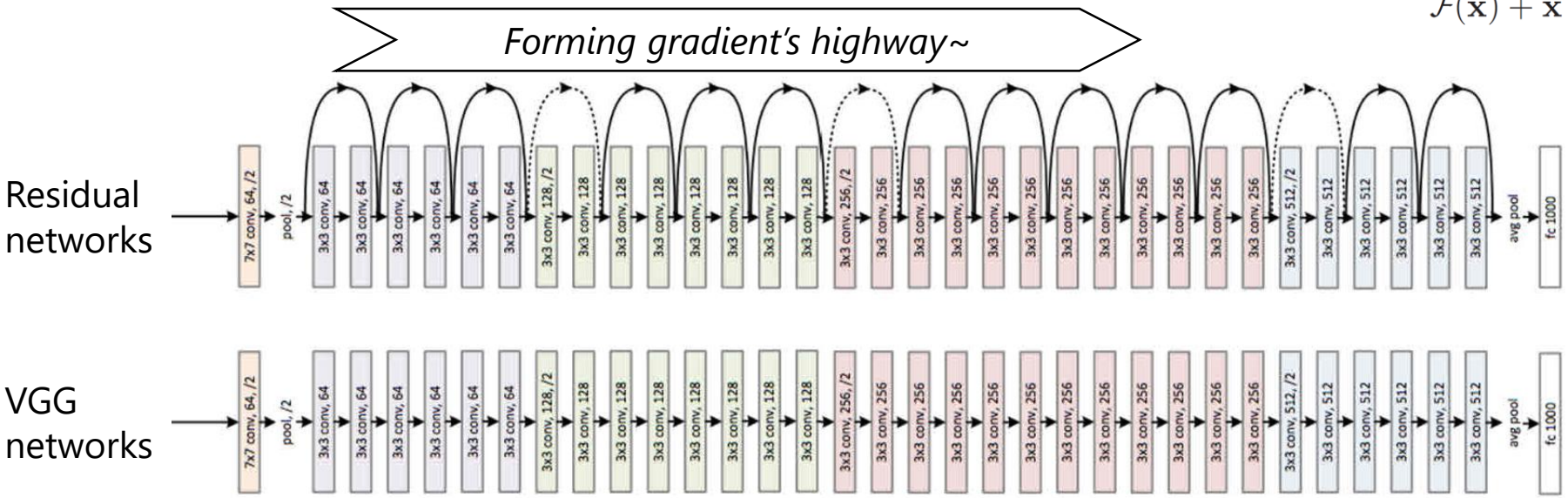
ResNet: ImageNet Challenge (2015) Winner

Major breakthrough in the network architecture

→ Better than "human performance" in ImageNet Challenge

→ **Residual units** make models easy to train!





(# cites: **112,800** in Mar. 2022)

[1] He, Kaiming, et al. "Deep residual learning for image recognition." CVPR 2016.

Summary

Basic computer vision tasks inspired by human visual system

→ Semantic and geometric scene understanding

→ Basic visual perception tasks: image classification, detection, segmentation, …

Basic deep neural networks

→ AlexNet, VGG, GoogleNet, ResNet, …

→ Take-home message:

"Not all **complex** and **deep** networks are good, but how well you **regularize** multi-dimensional features is the key to improve the performance."

KENTECH
Korea Institute of Energy Technology

Week 07a – 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.

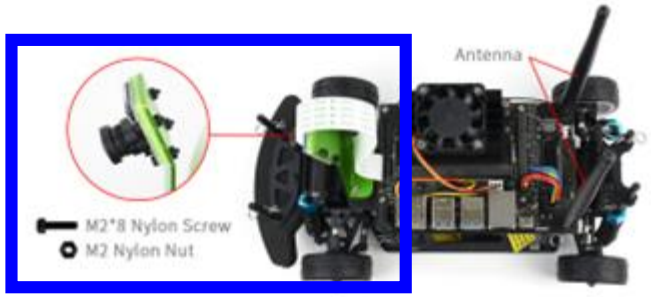
3. Fix JetRacer Pro Expansion board on chassis.

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**!

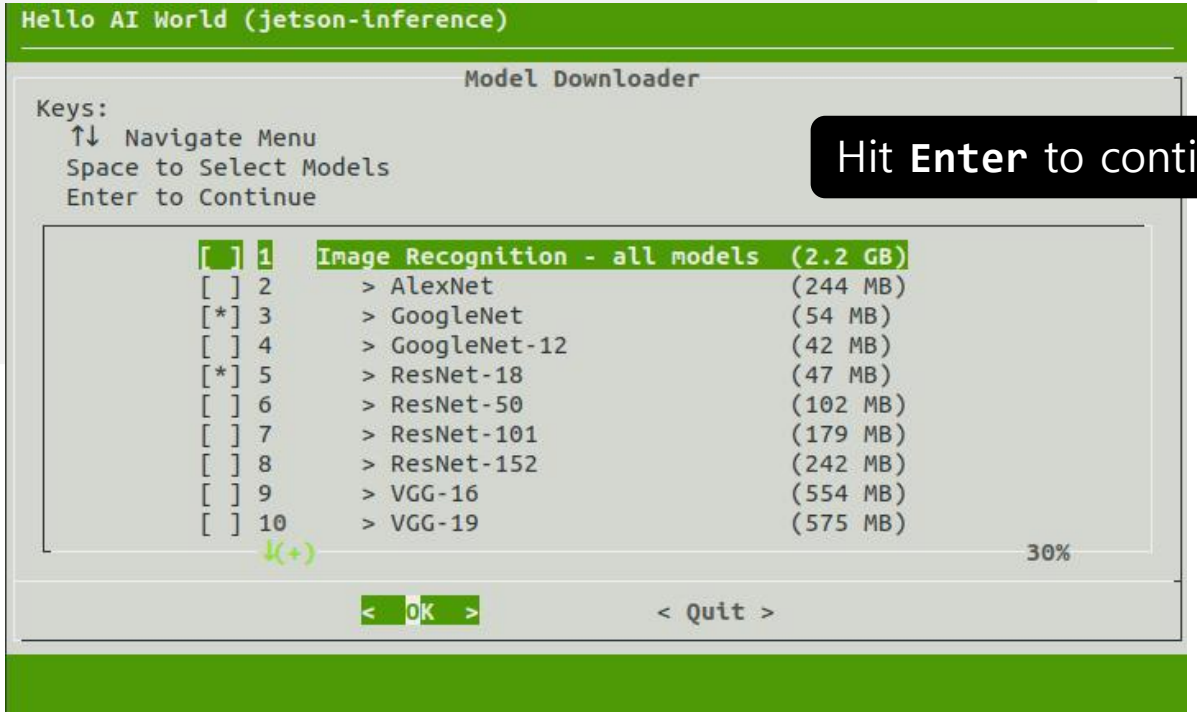
Configurations: Basic Packages

1. Open a browser and navigate to `192.168.***.***:8888`
2. Sign in with default password `jetson`
3. Open a terminal in Jupyter Lab by click `File → New → Terminal`
4. Follow **Step 5.** in https://www.waveshare.com/wiki/JetRacer_Pro_AI_Kit
   - Update Jetcard Python
   - Install JetCam Python package
   - Install torch2trt Python package
   - Install JetRacer package

Configurations: Hello-AI-World by NVIDIA

1. 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 ../
$ make -j$(nproc)
$ sudo make install
$ sudo ldconfig
```

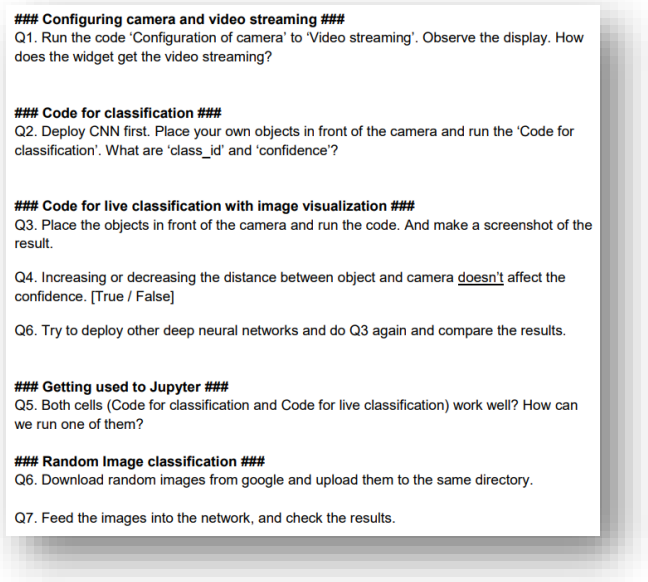


Hit **Enter** to continue!

You don't need to install PyTorch!

Run the Code JupyterLab

1. Copy and run **`do_classification.ipynb`** in **`~/jetson-inference/python/examples`**
2. This is a live demo for image classification.
3. Try to classify your own objects!
4. Try to classify random images downloaded from Google!
5. Please discuss each step with your teammate.
   - Example discussions in LMS



Configuring camera and video streaming
Q1. Run the code 'Configuration of camera' to 'Video streaming'. Observe the display. How does the widget get the video streaming?

Code for classification
Q2. Deploy CNN first. Place your own objects in front of the camera and run the 'Code for classification'. What are 'class_id' and 'confidence'?

Code for live classification with image visualization
Q3. Place the objects in front of the camera and run the code. And make a screenshot of the result.

Q4. Increasing or decreasing the distance between object and camera doesn't affect the confidence. [True / False]

Q6. Try to deploy other deep neural networks and do Q3 again and compare the results.

Getting used to Jupyter
Q5. Both cells (Code for classification and Code for live classification) work well? How can we run one of them?

Random Image classification
Q6. Download random images from google and upload them to the same directory.

Q7. Feed the images into the network, and check the results.