KENTECH
Korea Institute of Energy Technology

Visionary Course – Energy AI
Week 09

Seokju Lee

KENTECH
Korea Institute of Energy Technology

Introduction to Autonomous Driving

What You Will Learn…

1. Overview of Autonomous Driving

- Key components for self-driving
- Challenges and limitations

2. Overview of Computer Vision

- Basics of computer vision
- Visual perception tasks

3. Playing with Edge AI device

- Getting used to Linux system
- AI programming on NVIDIA Jetson
- Experiencing computer vision tasks

What is Autonomous Driving?

Driving without human intervention

Requirement: human intelligence

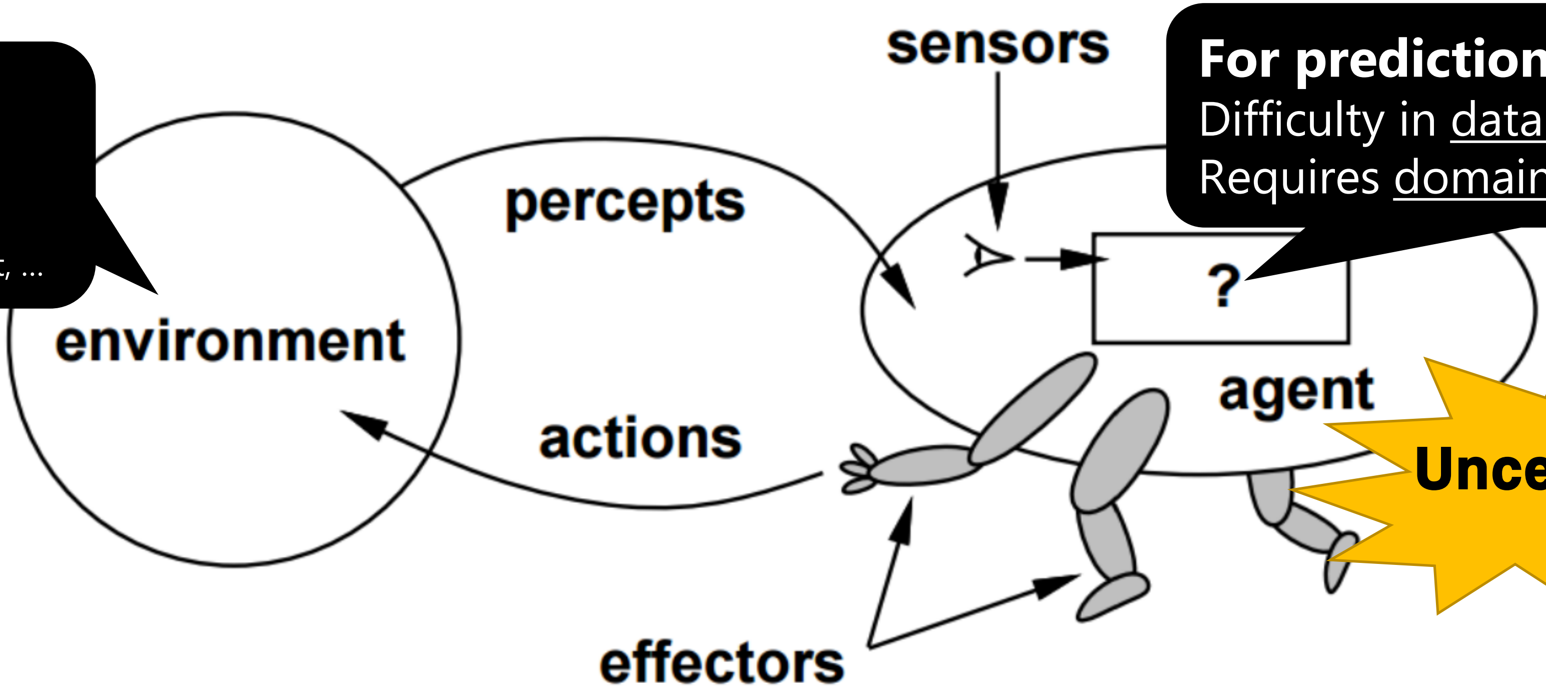
How to make it possible?

Can artificial intelligence (AI) achieve this?

What is AI? Should it be similar to human intelligence..?

Review: AI as a System

In real world,
Super high dimensional data!
e.g., image, video, sound, text, …



For prediction,
Difficulty in data representation.
Requires domain-specific knowledge.

Uncertainty!

Uncertainty ↓

Special-purpose AI:
Can it achieve a well-defined finite set of goals?

General-purpose AI:
Can it achieve poorly-defined unconstrained set of goals?

More uncertainty ↑

Q. Can we get closer to the general-purpose AI through bunches of special purposes?

AI for Different Levels of Tasks






Formal tasks:
Playing board games, solving puzzles, mathematical problems

Expert tasks:
Medical diagnosis, engineering, teaching, programming

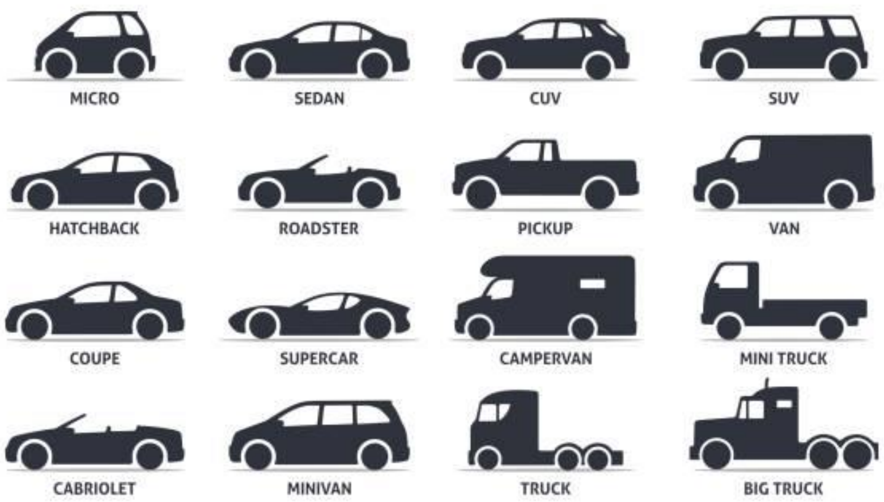
Mundane tasks:
Everyday conversation, walking, perception, dreaming in a sleep

How Difficult is Driving?

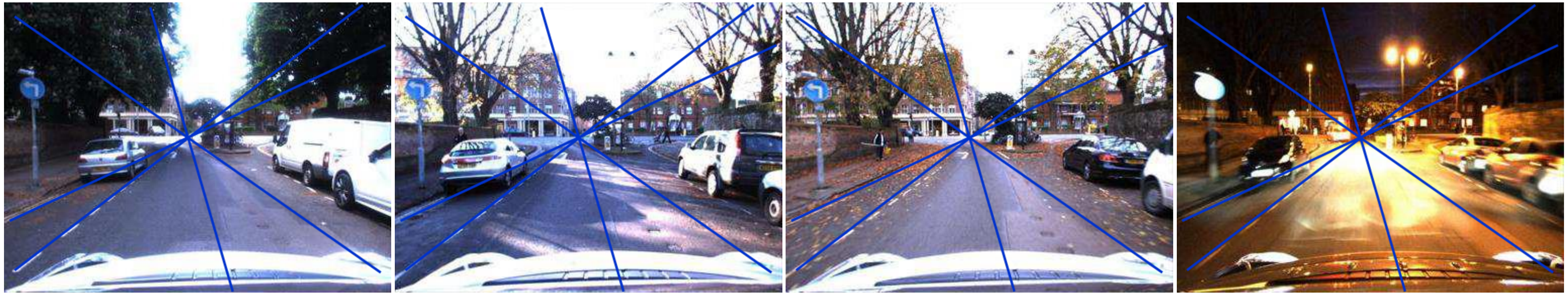
Is driving closer to **chess** or **everyday conversation**?

Why is Driving Difficult?

Different types of objects



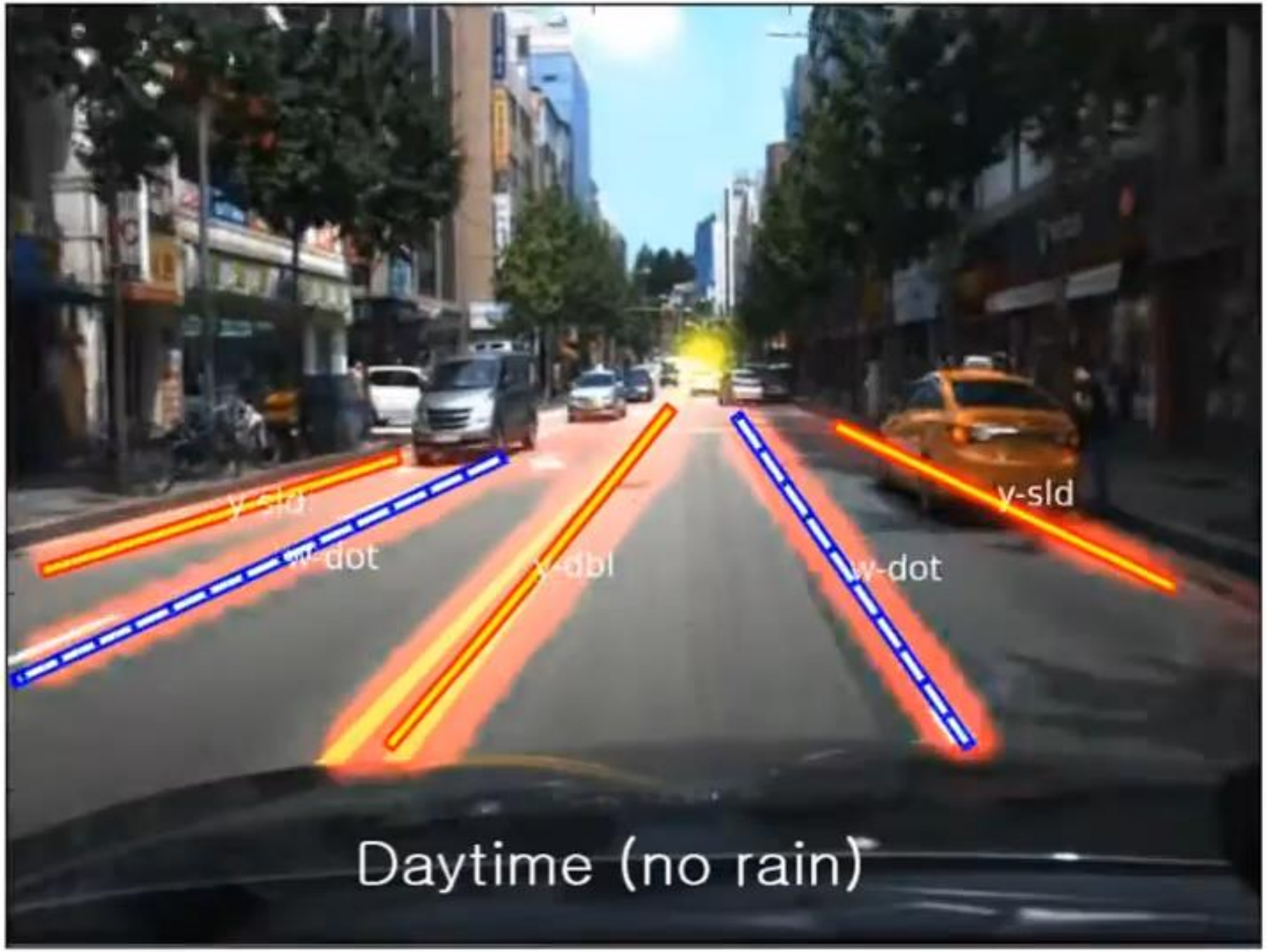
Dynamic environments



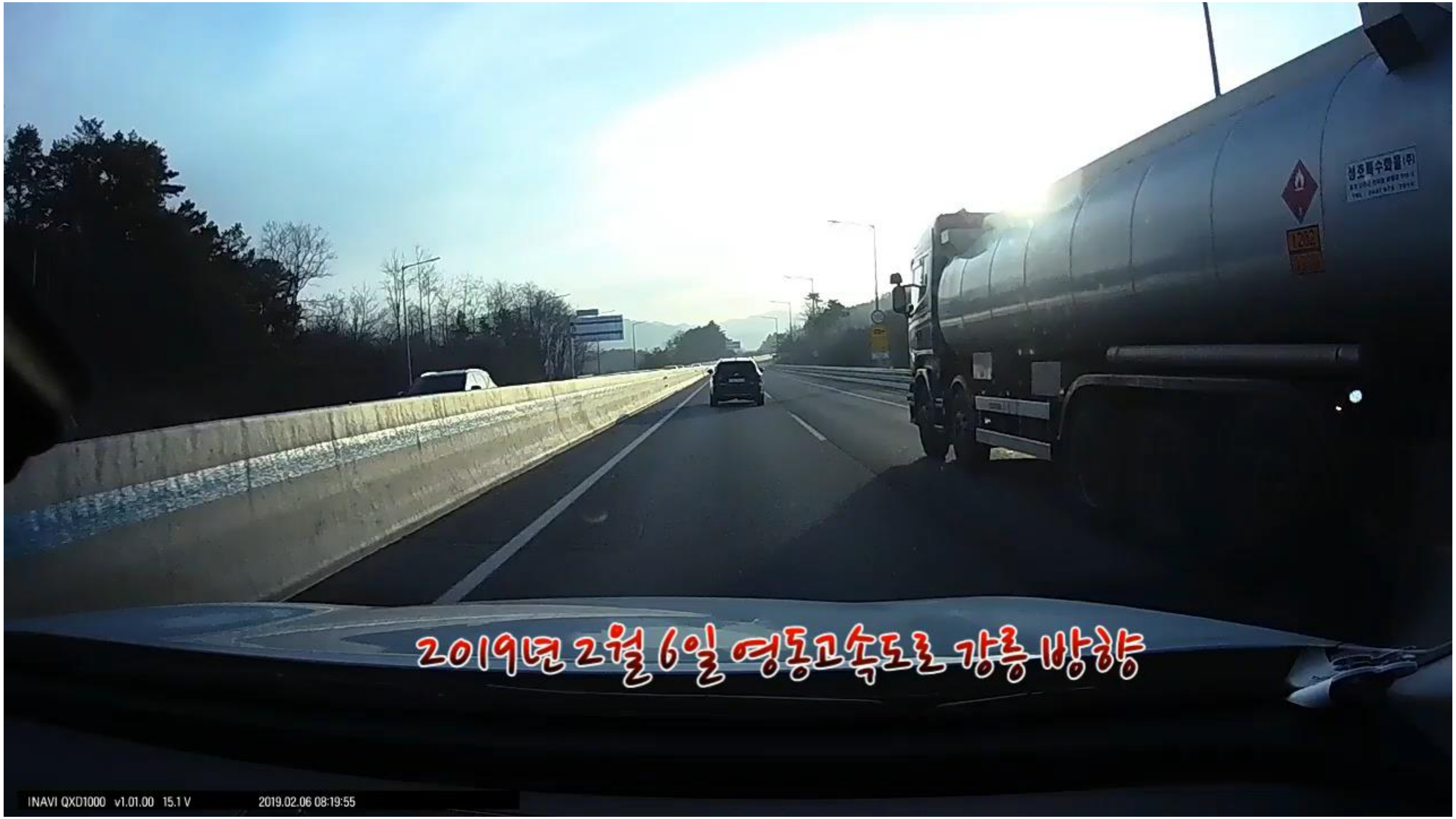
Same place, but
different weathers
(from RobotCar dataset [1])

[1] Maddern, Will, et al. "1 year, 1000 km: The Oxford RobotCar dataset." The International Journal of Robotics Research, 2017.

Why is Driving Difficult? (ICCV'17)



Why is Driving Difficult?



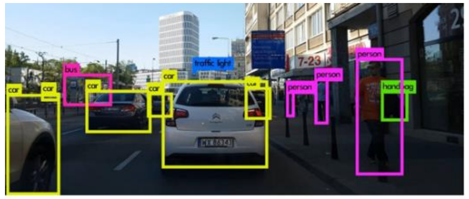
Three Pillars of Autonomous Driving

Sensing

Perceiving

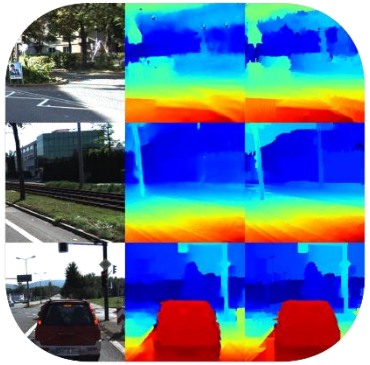
Control

Camera, RADAR, LiDAR, GPS, ultrasonic, stereo cam, surround-

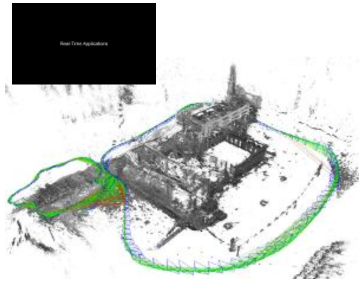


Object detection & recognition

Segmentation

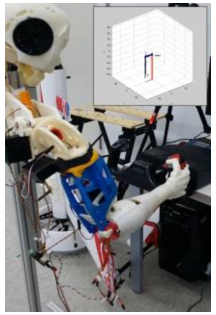


Depth



Localization and mapping

Motion/path planning, reinforcement learning, adaptive control, …



KENTECH
Korea Institute of Energy Technology

Understanding Robotic Perception

DARPA Grand Challenge II (2006)

[Stanford wins]

DARPA Urban Challenge (2007)

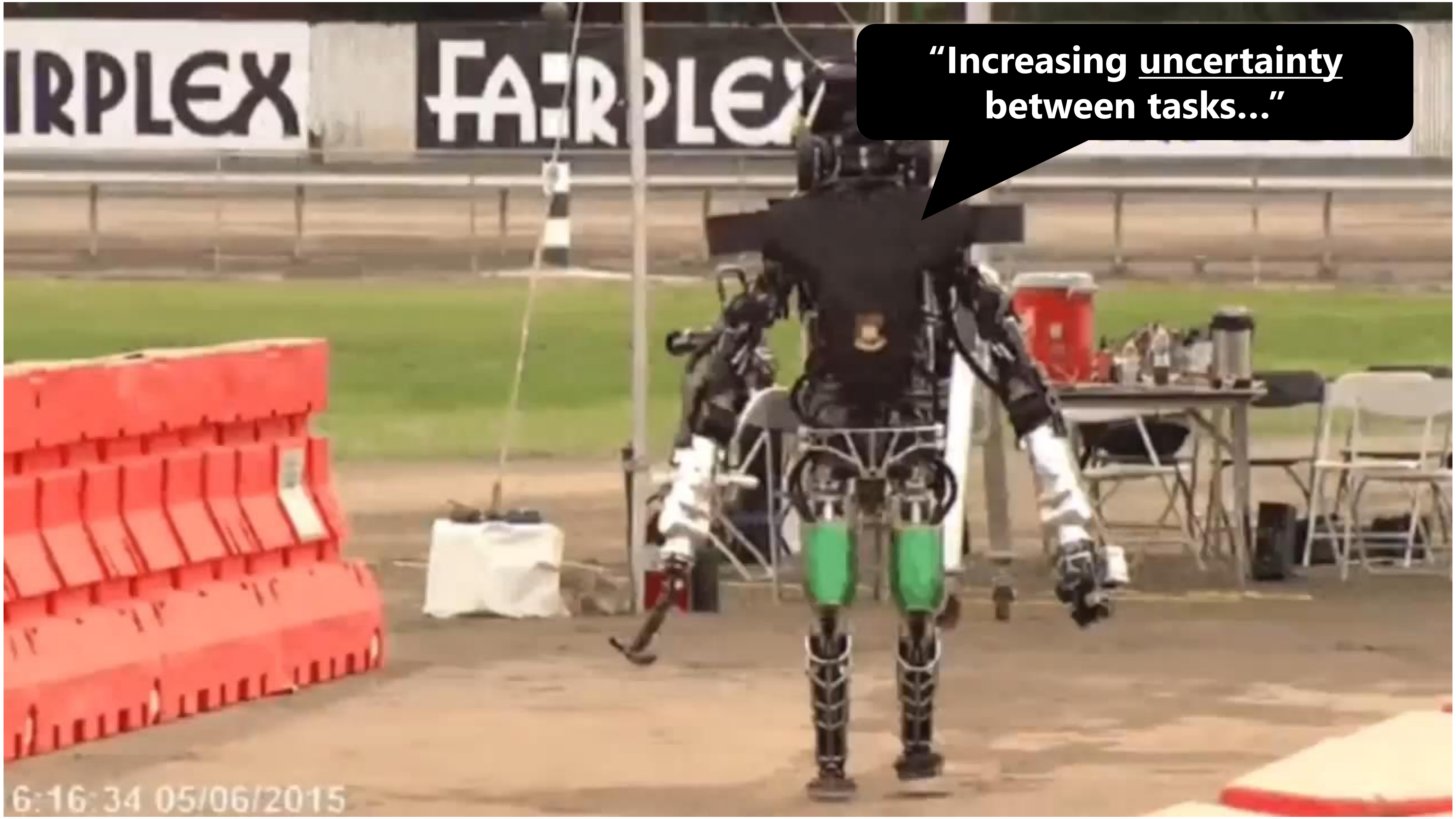
[CMU wins]

DARPA Robotics Challenge (2015)

KAIST wins

DARPA Robotics Challenge (2015)

Fails compilation



"Increasing uncertainty between tasks…"

NASA Sample Return Robot Challenge (2016)

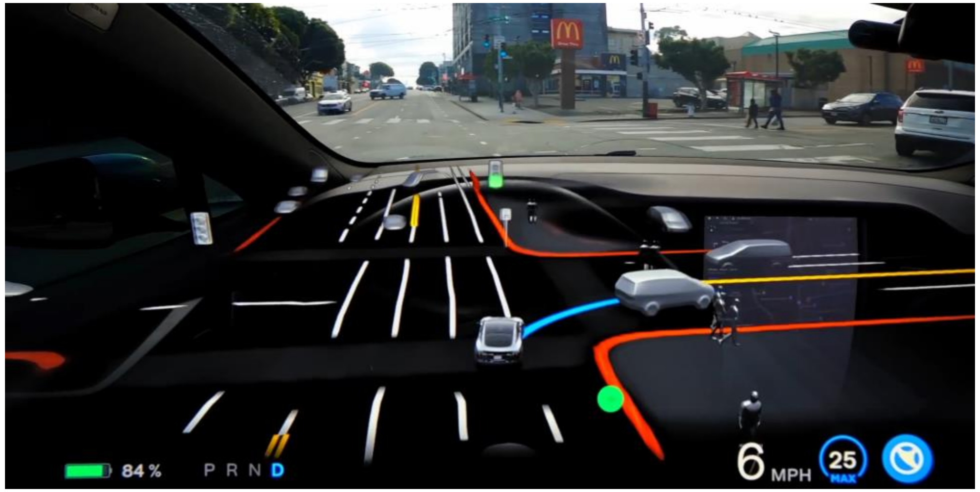
WVU wins

Indy Autonomous Challenge @ CES (2022)

Link

Industry Takes on the Challenge

Waymo’s self-driving taxis
(Dec. 2021)



Tesla’s full self-driving beta
(Jan. 2022)

“Accelerated by the recent breakthroughs in **deep learning**”

Q&A

KENTECH
Korea Institute of Energy Technology

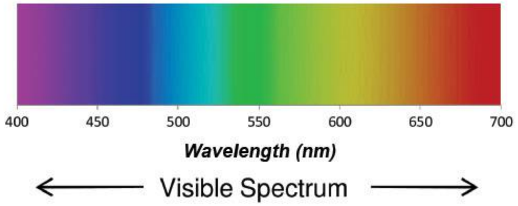
KENTECH
Korea Institute of Energy Technology

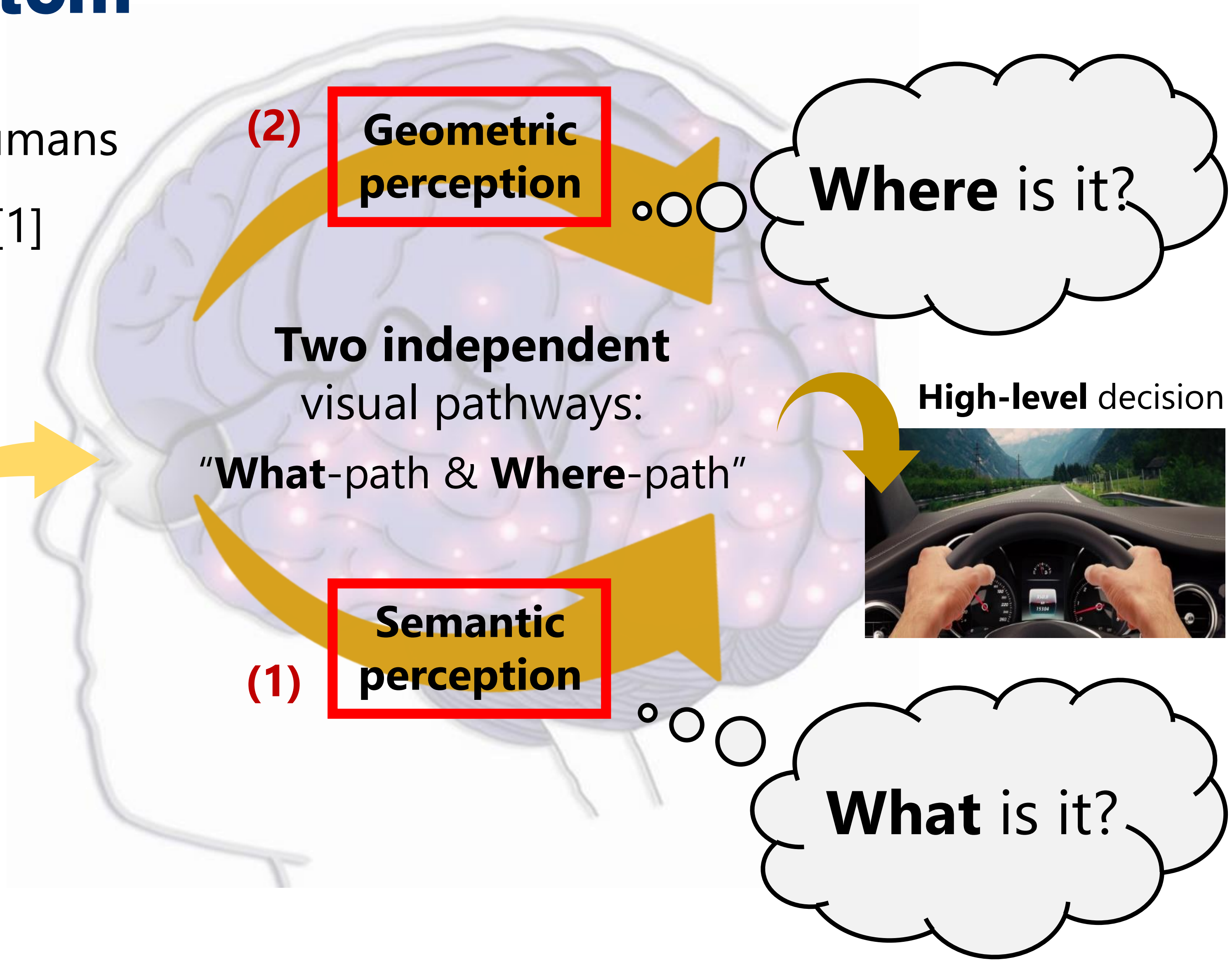
Computer Vision: Visual Perception

Human Visual System

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

Low-level visual signal





[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

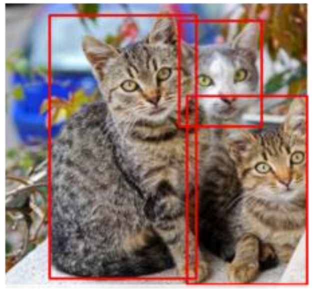
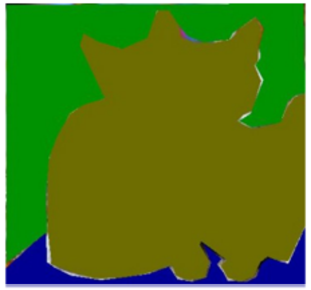
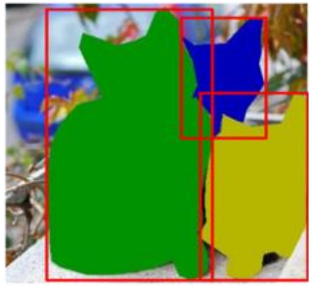
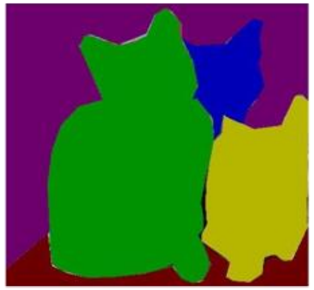
ex) 3D motion estimation

Computer Vision Tasks

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

Model Complexity ⇧ Output dimension ⇧

Image

Image Classification	Object Detection	Semantic Segmentation	Instance Segmentation	Panoptic Segmentation
				
"Image-level"	**"Box-level"**	**"Pixel-level"**	**"Pixel-level" + "Instance-level"**	**"Unknown pixel"**
	Fast RCNN, CVPR 2015 **Faster RCNN** NeurIPS 2015 **YOLO**, CVPR 2016 **SSD**, ECCV 2016 **Cascade RCNN**, CVPR 2017 **CornerNet**, ECCV 2018 **HTC**, CVPR 2019	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	DeepMask, NeurIPS 2015 MNC, CVPR 2016 FCIS, CVPR 2017 Cascade Mask-RCNN, PAMI 2019 **HTC**, CVPR2019 ⋮	
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

Video

KENTECH
Korea Institute of Energy Technology

Image Classification with Jetson

Image Classification

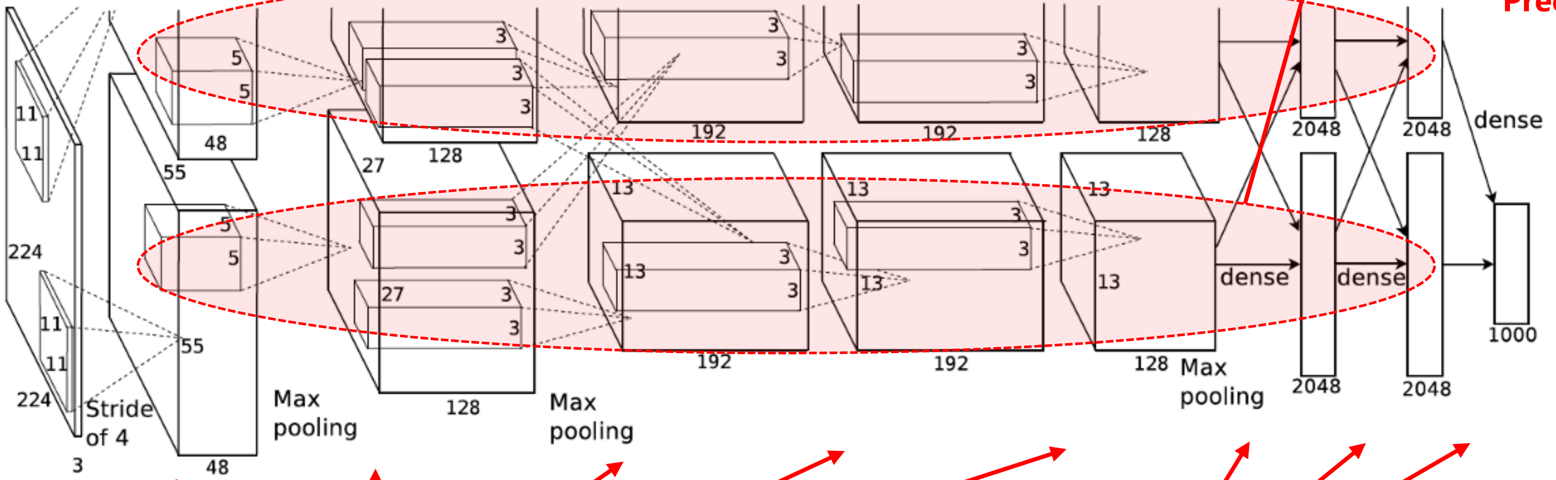
The most fundamental task for visual perception!

→ What's going on inside deep neural networks?

→ AlexNet [1] (# cites: **118,000** in Oct. 2022)

Memory ↑
Two GPU instances

Probability
(sum = 1)



Predicted class

→ Cat (0.94)
→ Dog (0.03)
→ Car (0.00)
..
→ Cup (0.01)
→ Bus (0.00)

Five convolutional layers
(feature extraction)

Three fully-connected layers
(classification)

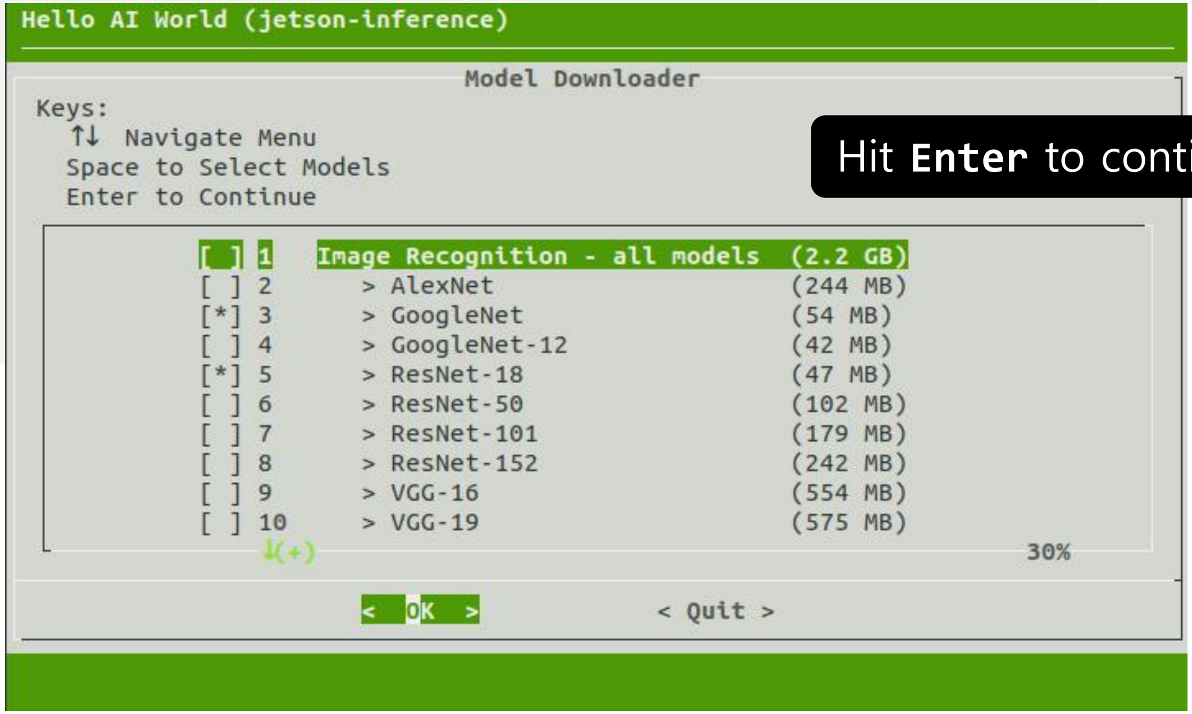
Predefined classes (*n*)
fixed size (=*n*) of array

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

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
```



Hit **Enter** to continue!

You don't need to install PyTorch!

Experiments (Report Due ~11/3)

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 (Report Due ~11/3)

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 (Report Due ~11/3)

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.

https://view.kentech.ac.kr

Q&A

KENTECH
Korea Institute of Energy Technology

Computer Vision Tasks

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