

# DL Lab #1:

## Object Detection using YOLO

Sunglok Choi, Assistant Professor, Ph.D.  
Computer Science and Engineering Department, SEOULTECH  
[sunglok@seoultech.ac.kr](mailto:sunglok@seoultech.ac.kr) | <https://mint-lab.github.io/>

# Overview

- **Prerequisite**

- Anacodna (Individual Edition) with PyTorch Installation
- Google Colab

- **Practice) Object Detection using YOLO**

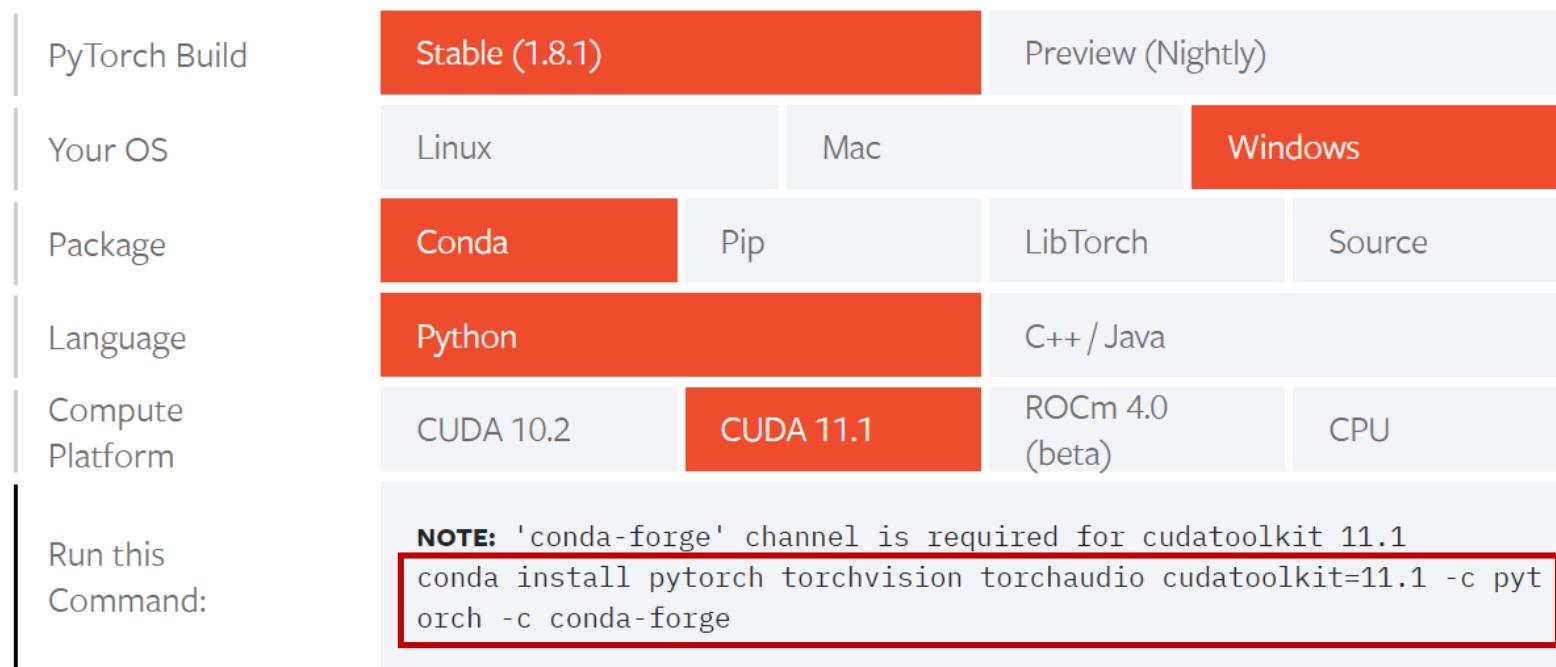
- The given data
- Expected results
- Practice with the skeleton code
  - Step #1) Run the given the skeleton code

- **Assignment**

- Mission: Run the given skeleton code

## Review) PyTorch Installation

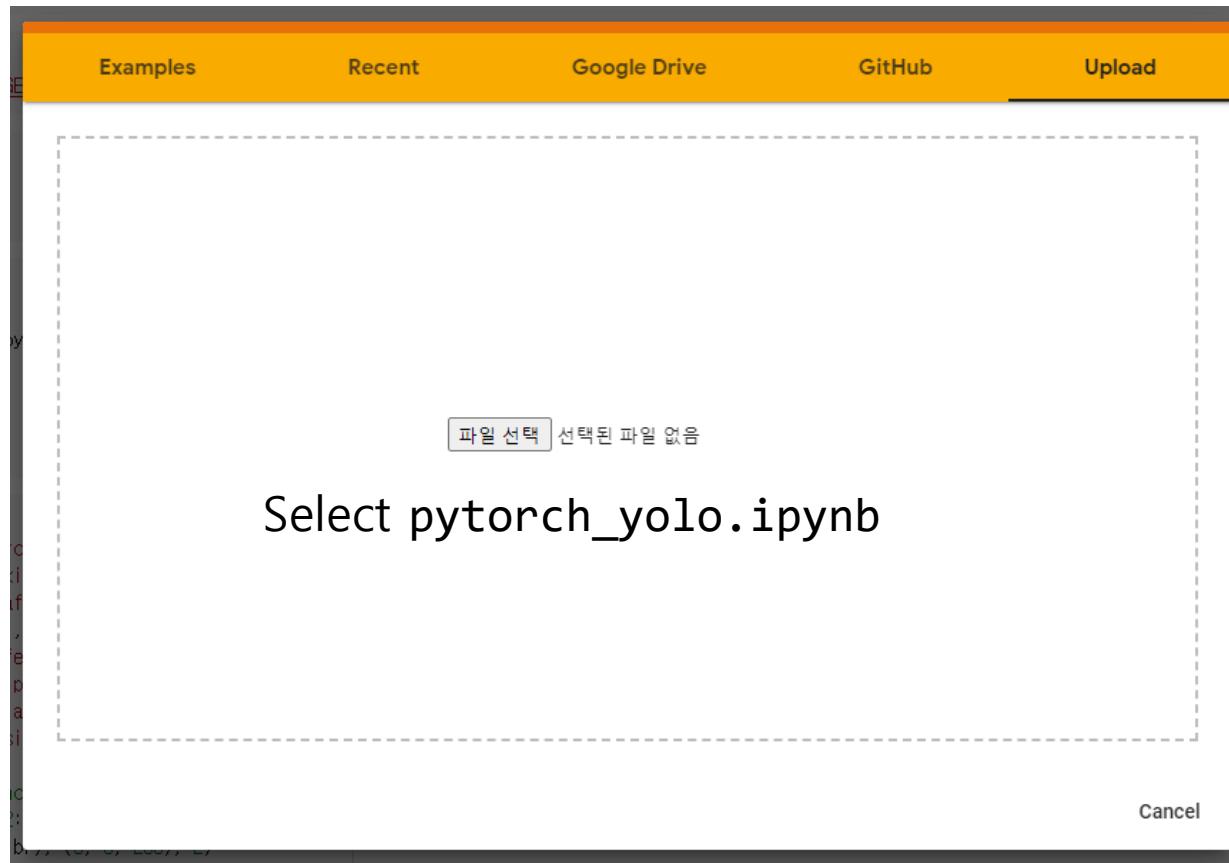
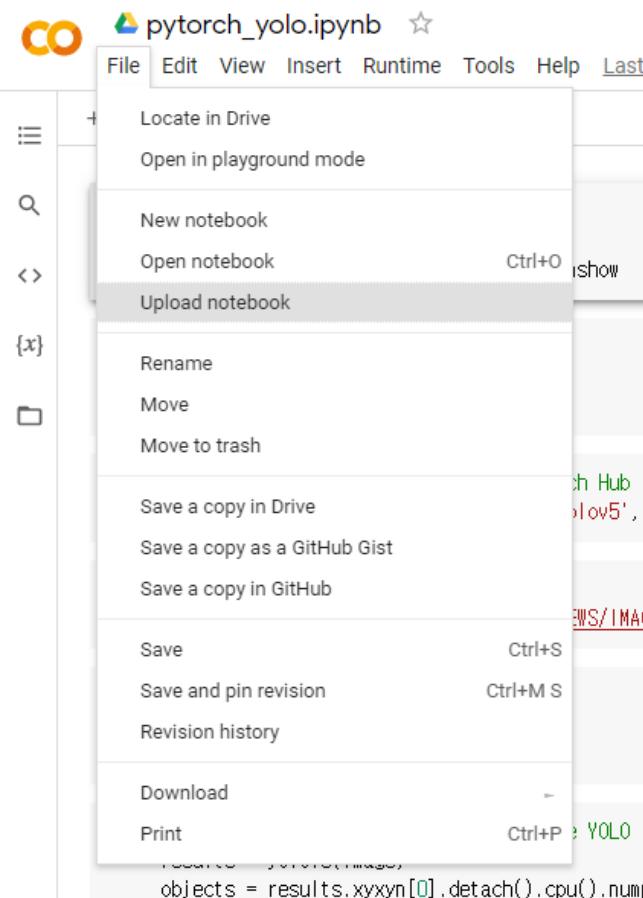
- Please follow [PyTorch's instruction of installation](#) for your system.
  - Note) If you want GPU acceleration, please install the matched version of CUDA in advance. Please visit [CUDA Toolkit Archive](#) to download a specific version of CUDA.



- Please use the given `pytorch_yolo.py` and `test.jpg` for the today's practice.

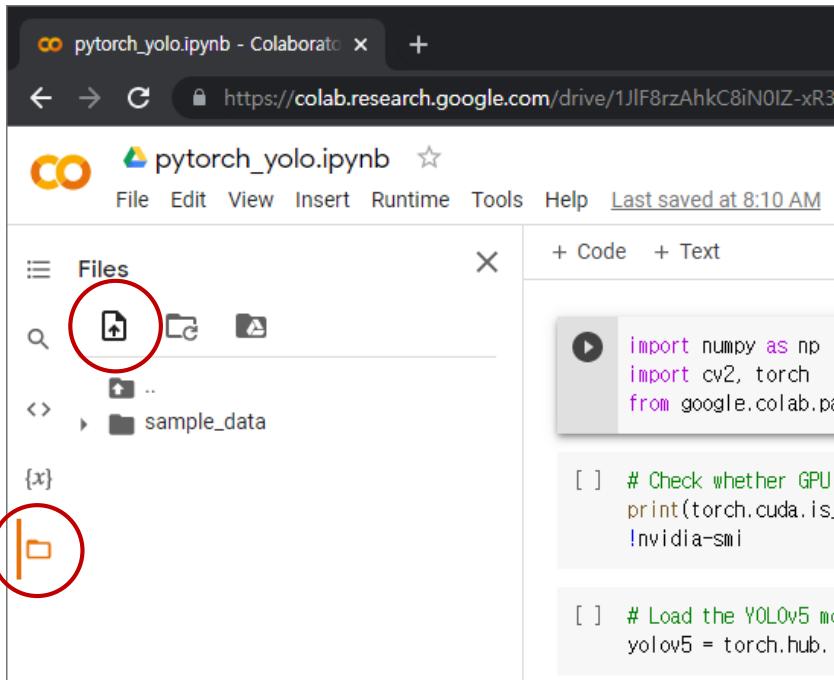
# Google Colab

- [Google Colaboratory](#)
  - It requires Google account.
    - Click "Sign in" at the top-right if you don't have or log in.
- Please upload the given notebook file, `pytorch_yolo.ipynb`.



## Practice) Object Detection using YOLO

- The given data (test.jpg)
  - Method #1) Upload the given data



A screenshot of a Google Colab notebook titled "pytorch\_yolo.ipynb". The left sidebar shows a file tree with a folder named "sample\_data". Two specific icons are highlighted with red circles: the "Upload" icon (a cloud with a plus sign) and the "Open" icon (an orange square with a white border). The main area contains a code cell with the following Python code:

```
import numpy as np
import cv2, torch
from google.colab.patches import install_ipython_colab
install_ipython_colab()
# Check whether GPU is available
print(torch.cuda.is_available())
# Load the YOLOv5 model
yolov5 = torch.hub.load('ultralytics/yolov5', 'custom', path='yolov5s.pt', source='local')
```



- Method #2) Download the image from internet

```
# Download an image from internet
!wget -c 'https://dimg.donga.com/wps/NEWS/IMAGE/2014/11/26/68179447.1.jpg' -O 'test.jpg'
```

# Practice) Object Detection using YOLO

- Expected results

The screenshot shows a Jupyter Notebook interface running on Google Colab. The notebook is titled "pytorch\_yolo.ipynb". A red circle highlights the "Runtime" tab in the top menu bar. Another red circle highlights the variable "classes" in the "Variables" panel, which contains a list of 80 object classes. The main code cell displays a script for visualizing detection results. The output cell shows a video frame with multiple objects detected by the YOLO model, each labeled with its name and confidence score. The objects include a person, a potted plant, a keyboard, a chair, a computer monitor, and a book.

```
# Show the image with results
classes = ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light',
           'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
           'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
           'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle',
           'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange',
           'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed',
           'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
           'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']

for obj in objects:
    if obj[-2] > 0.5: # More than 0.5 confidence
        t1, br = obj[0:2].astype('int'), obj[2:4].astype('int')
        cv2.rectangle(image, tuple(t1), tuple(br), (0, 0, 255), 2)
        cv2.putText(image, f'{classes[int(obj[-1])]}: {obj[-2]:.2f}', tuple(t1 + (-2, -4)), cv2.FONT_HERSHEY_DUPLEX, 0.4, (0, 0, 255))
cv2.imshow(image)
```

Output:

A video frame from a scene in a room. Several objects are detected and labeled with their names and confidence scores. The labels and scores are:

- potted plant: 0.79
- sofa: plant: 0.75
- sofa: plant: 0.80
- person: 0.86
- cup: 0.73
- cup: 0.74
- Keyboard: 0.74
- chair: 0.66

## Practice) Object Detection using YOLO

- The given skeleton code (1/2)

```
import numpy as np
import cv2, torch
from google.colab.patches import cv2_imshow

# Check whether GPU is available or not
print(torch.cuda.is_available())
!nvidia-smi

# Load the YOLOv5 model from the Pytorch Hub (https://pytorch.org/hub/)
yolov5 = torch.hub.load('ultralytics/yolov5', 'yolov5l', pretrained=True)

# Download an image from internet
!wget -c 'https://dimg.donga.com/wps/NEWS/IMAGE/2014/11/26/68179447.1.jpg' -O 'test.jpg'

# Load an image on internet
image = cv2.imread('test.jpg')
cv2_imshow(image)
```



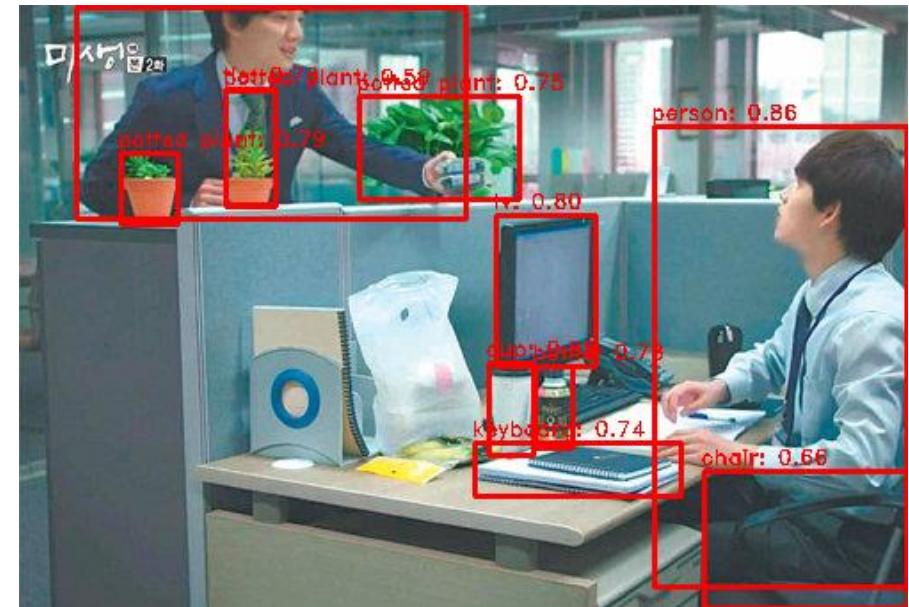
## Practice) Object Detection using YOLO

- The given skeleton code (2/2)

```
# Detect objects on the image using the YOLO
results = yolov5(image)
objects = results.xyxy[0].detach().cpu().numpy()

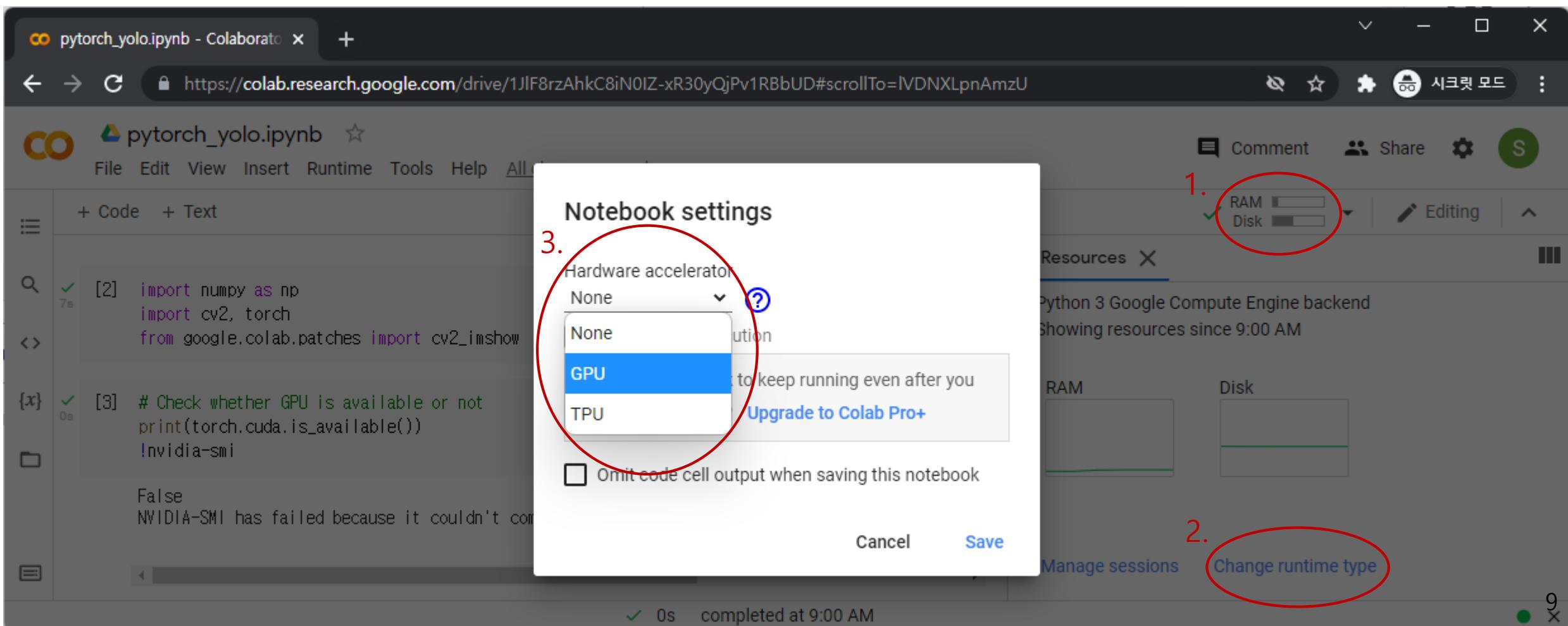
# Rescale object locations
h, w, _ = image.shape
objects[:,0:4] = objects[:,0:4] * [w, h, w, h]

# Show the image with results
classes = [ 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', ... ]
for obj in objects:
    if obj[-2] > 0.5: # More than 0.5 confidence
        tl, br = obj[0:2].astype('int'), obj[2:4].astype('int')
        cv2.rectangle(image, tuple(tl), tuple(br), (0, 0, 255), 2)
        cv2.putText(image, f'{classes[int(obj[-1])]}: {obj[-2]:.2f}', tuple(tl + (-2, -4)), cv2.FONT_HERSHEY_DUPLEX, 0.4, (0, 0, 255))
cv2_imshow(image)
```



## Tip) If Your Session does not have GPU

1. Click "RAM / Disk" (or list box(▼) > View resources)
2. Click "Change runtime type"
3. Select "Hardware accelerator" as "GPU"



# Assignment

- Mission
  - Run the skeleton code with your desired image (or video)
  - Submit your screenshot (screenshot.png) on your web browser or Anaconda
- Condition
  - You can start from scratch (without using the given skeleton code).
    - However, you should use another image or video.
  - You can freely change the given skeleton code if necessary.
- Submission
  - Deadline: **November 27, 2024 23:59** (firm deadline; no extension)
  - Where: e-Class > Assignments
  - Score: Max 10 points