# Robotics finals project report

# Group members: Pham Xuan Thanh – 16022419

Nguyen Duc Thao - 16020283

Phan Dang Trung Hieu - 16020227

# I. Requirements

1. System:

- OS: Mac OSX 10.9+, Ubuntu 14.04+.

- Graphics Card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.

- CPU: SSE2 instruction set support.

- Python 3.6+

- Linux: X server with GLX module enabled

2. Pip requirements

- Matplotlib

- OpenCV

- pynput

- pillow

- Ai2thor

- keyboard

- virtualenv (optional)

# II. Installation

**1. Get repository:**

git clone: https://github.com/pxthanh98/robotics.git

**2. Run code:**

- Challenge 1: python challenge\ 1/main.py

- Challenge 2: python challenge\ 2/trajectory.py

# III. Solution

**1. Challenge 1:**

- YOLO (You Only Look Once) is a method / way to do object detection. It is the algorithm /strategy behind how the code is going to detect objects in the image. In this challenge, we are going to use a pre-trained YOLO model with OpenCV to detecting objects [1]

- DNN (Deep Neural Network) module was initially part of opencv\_contrib repo. It has been moved to the master branch of opencv repo last year, giving users the ability to run inference on pre-trained deep learning models within OpenCV itself.

- For this challenge, we use pre-trained weights file named yolov3-thor\_final.weights which was trained by YOLO network with Ai2thor objects dataset: <https://drive.google.com/file/d/1nks4PxeFBiiSZP-KQCi9U0WPRt_Gk6Ut/view?usp=sharing>

- “main.py” file took the responsiblity for nteraction between agent and virtual environment when “detection.py” file took the responsibility for object detection

- Agent will take a picture represent the scene of environment and every objects in it, write it to “image.jpg”.



Figure 1. image.jpg - The image captured by agent

Then using the image to get prediction “prediction.jpg”



Figure 2. prediction.jpg – Prediction of agent

**2. Challenge 2:**

- For the second challenge, we treated it like a simple navigation problem and record every stage of the agent in ai2thor environment everytime a key is pressed

- This challenge can be procceed by 3 steps: Get view, Localization, Processing.

- Get view: At this step, we need a different point of view of agent to able to obtain the whole room. To do that we used function:

**controller.step(dict(action=‘ToggleMapView’))**

- Localization: We record agent’s position at every stage, convert it into a 2D frame for processing using a **converter(object)** [2]

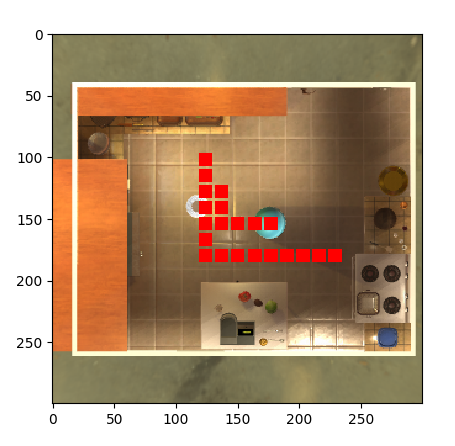
- Processing: Interaction between agent and ai2thor environment, draw the stage of agent with matplotlib.

Figure 3. Agent’s initial position

Figure 4. Agent’s trajectory

# IV. References

[1] <https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9>

[2] <https://github.com/allenai/ai2thor/issues/124>