ROS2 Computer Vision

Learning outcome

This exercise will help students to familiarize with the implementation of basic computer vision tasks such as object detection and tracking in both simulation and real environment. The simulation task is associated with ROS and Ignition Gazebo Simulator. The vision algorithms are implemented in python programming language. Anaconda python distribution is used for managing python environments and packages required for the exercise. Finally, jupyter notebooks will be used for a more interactive and understandable presentation.

Grading

Grade 3: task 1 + 2 + 3, successful demo and could answer all questions
Grade 4: task 1 + 2 + 3 + 4, successful demo and could answer main questions
Grade 5: task 1 + 2 + 3 + 4 + 5, successful demo and could answer main questions

Equipment:

This exercise requires Ubuntu 20.04 and ROS foxy and a working usb/webcam device.

TASK 1: Build the ROS Package

In this task you will be simulating a walking actor and a camera sensor. The camera is pointed towards the walking actor and it captures the complete motion within its field of view. The source code and the instructions required to build the package is hosted in the following github repository. (switch to noetic branch)
https://github.com/KulunuOS/AUT.700-E4.git

Task 2: Installing Anaconda and setting up environment

If you haven't already installed Anaconda distribution in your operating system, follow the steps from the beginning. If not skip to step IV.
Download the latest Anaconda installer for Linux from here
Open a fresh terminal and enter the following command and install Anaconda for python 3

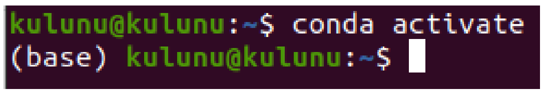
```
$ bash ~/ Downloads/Anaconda3-2020.02-Linux-x86_64.sh
```

In order to initialize after the installation process is done, first run

```
$ source <path to conda>/bin/activate
```

Then run $ conda init and $ conda activate sequentially

If you have installed anaconda properly, then the terminal will activate conda as below. If not please refer anaconda documentation here



```
kulunu@kulunu:~$ conda activate
(base) kulunu@kulunu:~$ 
```

Please read and understand the purpose of conda environments, If you are not already familiar with conda environments.

Create a conda environment with the environment.yml file provided with the repository.
Now you have finished creating the conda environment required to complete this exercise.

```
$  cd ~/ws/src/Notebooks
$ conda env create -f environment.yml
```

Activate the conda environment you just created by typing

```
$ conda activate tracker
```

To view the list of packages installed to your conda environment type the following

```
$ conda list
```

If you want to learn more about managing conda environments please follow the documentation here

Task 3: Object Detection and Tracking

In this task, you will be implementing an object detection and tracking algorithm in a Jupyter notebook environment. You will be introduced to OpenCV library which is widely used in computer vision tasks, some utility libraries used in image processing and performing computer vision tasks with ROS.

Open a new terminal and run the commands below

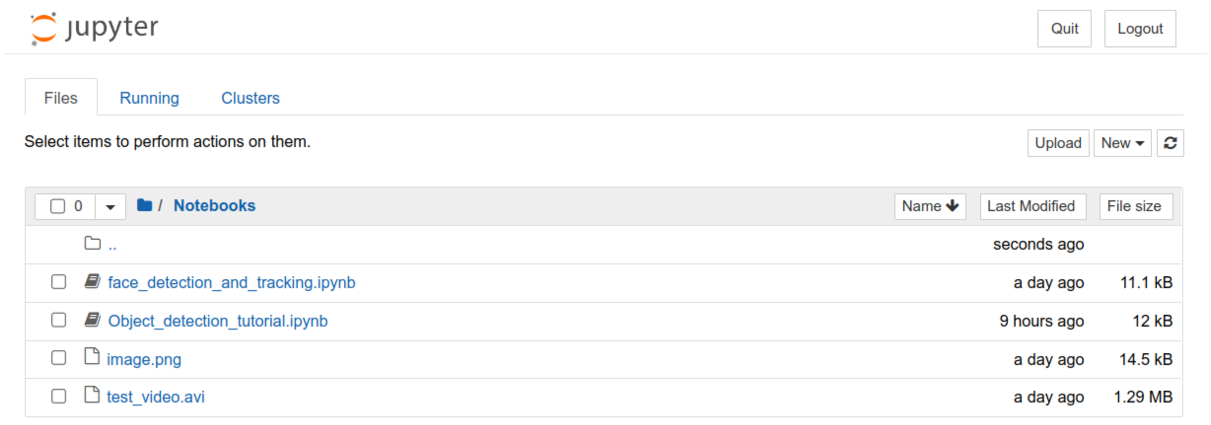
```
$ cd ~/ws/src/Notebooks
$ source /opt/ros/noetic/setup.bash
```

Jupyter-notebook comes pre-installed with anaconda. **Open a new terminal** and run the following command (in your conda base environment).

```
$ conda activate
$ jupyter notebook
```

The Jupyter-notebook will be opened in your browser.

Open Object_detection_tutorial.ipynb in your Jupyter notebook.

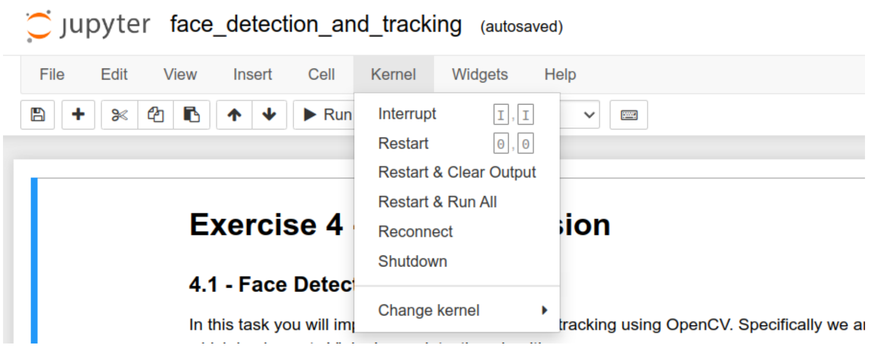


Select items to perform actions on them.

0 / Notebooks

..	seconds ago	
face_detection_and_tracking.ipynb	a day ago	11.1 kB
Object_detection_tutorial.ipynb	9 hours ago	12 kB
image.png	a day ago	14.5 kB
test_video.avi	a day ago	1.29 MB

Have a look at the kernels available in the notebook (kernel > change kernel). You will realize only python3 kernel is available.



However, we need to configure the notebook to run in the “tracker” conda environment we created. **Switch back to the previous terminal** where you activated the conda environment and run the following commands

```
$ ipython kernel install --user --name=tracker
```

Now you can see the new kernel in your notebook in top menu bar Kernel > change Kernel > tracker (refresh the notebook)
Complete the task guided as guided in the notebook. Remember to compile every code using the kernel we create – “tracker” . This allows us to import the custom modules that are required to run the notebook.

Task 4: Face Detection and Tracking

In this exercise you will implement a face detection algorithm using OpenCV. You will use Viola-Jones detection algorithm to find a face in an image and then forward to the KLT algorithm that follows it. The Kanade-Lucas-Tomasi (KLT) algorithm is a feature tracker, which first detects a set of feature points and then tracks these using optical flow.
Open face_detection_and_tracking.ipynb in the same folder as task 3. Switch the kernel to “tracker” and follow as instructed in the notebook.

Task 5: use input from webcam

Implement the above face detection and tracking on yours or your friend’s faces using an input from a webcam/usbcam and ROS.

Face redetection:

The face detector might detect a face in a place where no face is actual present or lose its track halfway the video. In the worst case, the point tracker tracks stationary points somewhere in background. This means that the tracker never “releases” the points, and the actual face is left unnoticed. To prevent this, one way is to perform a new face detection regularly (e.g. every 100 frames). Try to Implement this periodic face detection and improve tracking.

To arrange a demo contact: Kulunu (kulunu.samarawickrama@tuni.fi)