NATIONAL UNIVERSITY OF SINGAPORE

EE4705: Human and Robot Interaction

Project 1: Detection of human gestures and behaviors using PyTorch (15%)

I. Instructions:

1. This project is a joint **work to show your collaborative effort, and at the same time each individual can show his/her own talent for individual portion as your choices and declaration**. Students are required to work together to complete it with **3 people as a group** for the project manual with one example and five tasks.
2. Only one combined report should be submitted per group. Put your name and matriculation number on the cover page. Individual contributions and efforts are required to state clearly on Page 2. Detailed descriptions, explanations and interpretation should be presented for every point in each tasks.
3. Recommended pages: 5 – 15 pages, 1.5-line spacing, 12-point 'Times New Roman' font, and 1-inch margins
4. The detailed distribution of scores will be given in the following part.The most important objective of this project is to understand and grasp how to do a gesture classification task based on neural networks and implement it with PyTorch. The accuracy of your model would not take the big role in the grading (Considering the different types of CPU or GPU would give results with large difference).
5. Zip all files into one compressed file (including the report in PDF, the '*.ipynb'* files for five tasks, your own built dataset and other necessary data, and proper citations for all sources.). Name your file as the registration number of the first student per group. Submit the project into the folder at Canvas/Assignments/Project1 **before Sep. 22, 2023**

   *Example: If the first student's registration number is A*******X and then the file name should be Group-A*******X.zip.*
6. GPU is not compulsory for this project.
7. Any queries, feel free to contact GA Leng Yunze, E1143247@u.nus.edu

II. Score distributions

The distribution of the scores for each item is listed below. If an item is completed by more than one person, they will share the marks of the item equally.

Groups with 3 members

Task	Item	Score Distribution	Name
Task 0	1,2,3	10%	All
Task 1	1	10%	A
	2	15%	All
	3		
Task 2	1	10%	B
	2	20%	All
	3		
Task 3	1	10%	C
	2	25%	All
	3		
	4		
Task 4	1,2,3,4	20%	All

Groups with 2 members (in special case)

Task	Item	Score Distribution	Name
Task 0	1,2,3	10%	All
Task 1	1	10%	A
	2	30%	All
	3		
Task 2	1	10%	B
	2	30%	All
	3		
Task 4	1,2,3,4	20%	All

III. Objective

Gesturing is a natural and intuitive way to interact with people and the environment. Gesture recognition is an active research field in human-robot interaction. From teaching perspective, we adopt the most basic human gesture and behavior classification task as the theme of this project to stimulate students' interest in this research. The necessary guidance will be given in this manual. Laboratory assistants will also be available to provide limited guidance.

The objectives of this project are:

1. To familiarize with the gesture recognition and classification task using the data-driven method.
2. To provide hands-on experience in the acquisition and processing of gueture data from computer-based data acquisition devices.
3. To familiarize with the popular deep learning platform PyTorch.
4. To learn the entire process of developing a neural network based model, including preparing data, building neural network model, training, testing and visualization.
5. To explore the performance of different neural network models on the gesture classification.
6. To provide practical experience on problem-finding and problem-solving when doing deep learning based experiments.

IV. Example

Gesture classification experiment in this project is planned to execute based on deep learning method. Therefore, a simple example is here given to guide you to learn the whole process of building a classifier based on the neural network.

Please find the '*Example.ipynb*' file to see the detailed information.

V. Understanding body language by gesture recognition

The dataset used in the gesture classification project has been given to you, which includes 4 classes, named '*left, palm, peace and right*'. Please see the folder '*dataset*' for detailed information.

The aim of the Task 0 is to learn how object detection could contribute to HRI. In this task, we will calculate the distance between human faces and the computer camera. The partial code has been given in *Task 0.ipynb.*

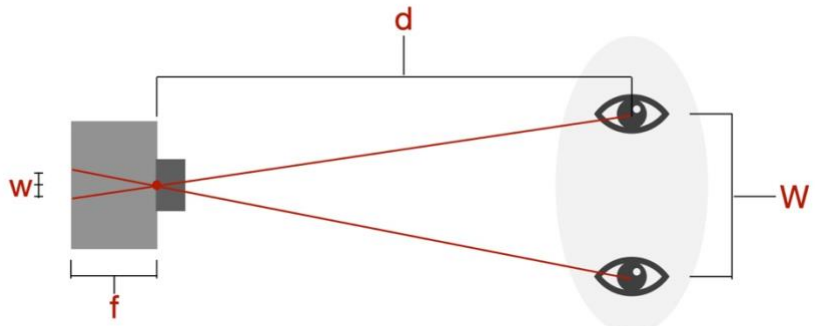
The aim of the Task 1, 2, and 3 are to learn how to build a model to classfy human gestures based on different neural networks (mainly fully connected neural network and Convolutional neural network). You will experience it in Task 1 and Task 2. Further more, to better understand this project, we will give a deep explore in Task 3. You are encouraged to collect the gestures or behaviors data and then to build your own dataset and model.

The aim of the Task 4 are to understand speech recognition and bulid a Speech-to-Text Conversion program. The input data should be the speech .mp3 file, and the output text should be printed.

The detailed requriments for these four tasks are illustrated in the corresponding *.ipynb* files (*Task 0.ipynb, Task 1.ipynb, Task 2.ipynb, Task 3.ipynb, and Task 4.ipynb*). Part of codes have been given in '*Task 0.ipynb*' *and* '*Task 1.ipynb*'. You are required to read them carefully and give your own results in right place. For Tasks 2 and 3, you need to code by yourself in files '*Task 2.ipynb*' and '*Task 3.ipynb*' referring to the codes in Task 1.

Task 0: Determine the distance between face and camera

1. Do a review on object detection and its application in Human-Robot Interaction. Learn the python library of cv2 and cvzone.
2. According to the existing open source library cvzone, build a facial recognition program to recognize faces from computer cameras and realize real-time distance measurement from faces to camera.
3. Run the model. Analyse and comment on the performance of your model, and discuss the importance of object detection in HRI.



f : camera focal length
w : the pixel width of the human eye after sensor imaging
W: the distance between the left eye and the right eye in reality
d: the distance between the human eye and the camera in reality

$$f = (w * d)/W \qquad d = (f * W)/w$$

Task 1: Understand body language by gesture recognition with fully connected neural network

1. Do a review on gesture recognition and its applications to in Human-Robot Interaction. Summarize what you have learned.
2. Referring to the previous example about building a neural network based classifier, use what you have learned to read the code for gesture classification and design your own network architecture using fully connected layers."
3. Run the model. Analyse and comment on the performance of your model based on fully connected layers.

Task 2: Understand body language by gesture recognition with convolutional neural network

1. Do a review on Convolution Neural Network. Learn how to build a convolutional layer in PyTorch.
2. Referring to the guide in Task 1, build your own network for gesture classification using convolutional layers. Please see the references 4 in this manual to learn how to build convolutional layers in PyTorch.
3. Analyse and comment on the performance of the model. Make a comparison between the fully connected based and convolutional based models and comment on it.

Task 3: Understand human gesture and body language based on your own built dataset and model

1. Do a review on dataset building and other deep learning based models applied on gesture recognition. Comment on their applications and benefits.
2. In the earlier two tasks, you have learned how to do the gesture classification task using the given dataset. Now, you need to collect data by yourself and build your own dataset. The dataset is not limited to gestures. **Body postures and behavior are encouraged**. Please place your data referring to the format of the given dataset. For good performance, the number of data in each class is recommended over 50. For the number of classes, it is better to have more than 3.
3. Design your own neural network architecture. Fully connected or convolutional layers used in the task 1 and 2 are acceptable. But you are encouraged to learn more deep learning models and

implement it as possible as you can."

4. Write down the problems you encountered during the experiment, the solutions, and your experiences.

Task 4: Understand speech recognition and speech-to-text conversion

1. Do a review on speech recognition and explain the principles or theory of voice singal processing.
2. Design a program to convert the given dataset_for_task4 MP3 file into text. Show your code and results in your report. Program can be done with open-source python libraries, e.g. “speech_recognition” or “pydub”.
3. Establish your own input dataset, and convert your own speech data into text. For example, by recording: “ I am having fun with human robot interaction class.” Show your code and results in your report.
4. Write down the problems you encountered during the experiment, the solutions, and your experiences.

VI. Toolkits

The tools that need to be installed are listed here. Please download and install them one by one.

1. **Install Python**
   - Download from the websites https://www.python.org/downloads/ and see this website to install it https://realpython.com/installing-python/
   - Version 3.9 is recommended
2. **Install PyTorch**
   - Download from the website https://pytorch.org/. Please notice to make your own choice according to your computer configuration

For Windows system,we recommend to install PyTorch under the configuration **Stabel, Windows, Pip, CPU**. Run the corresponding command in Windows PowerShell.

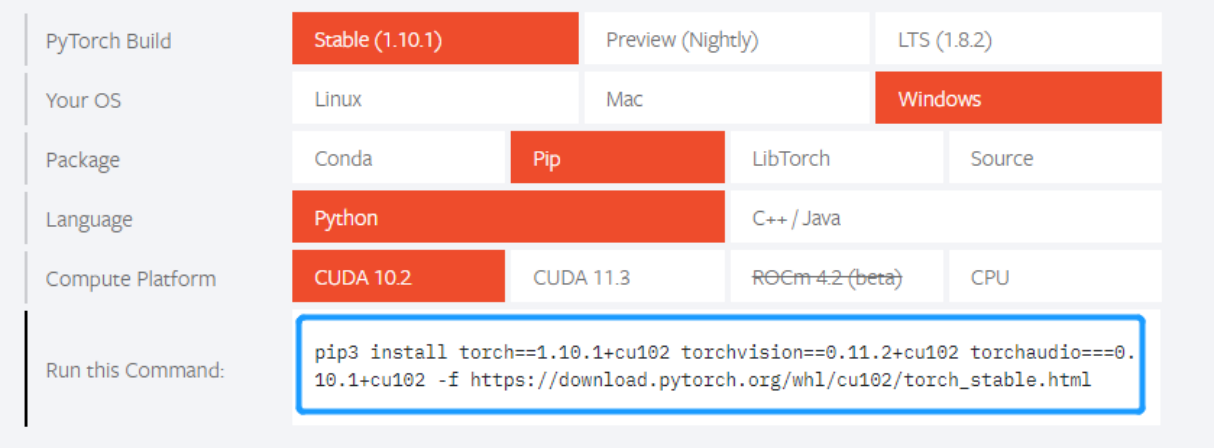
```
PS C:\WINDOWS\system32> pip3 install torch torchvision torchaudio
Collecting torch
  Downloading torch-1.10.1-cp39-cp39-win_amd64.whl (226.5 MB)
     |████████████████████████████████| 226.5 MB 6.4 MB/s
Collecting torchvision
  Downloading torchvision-0.11.2-cp39-cp39-win_amd64.whl (985 kB)
     |████████████████████████████████| 985 kB 6.8 MB/s
Collecting torchaudio
  Downloading torchaudio-0.10.1-cp39-cp39-win_amd64.whl (341 kB)
     |████████████████████████████████| 341 kB 6.4 MB/s
Collecting typing-extensions
  Downloading typing_extensions-4.0.1-py3-none-any.whl (22 kB)
Requirement already satisfied: numpy in d:\ruanjian\python\lib\site-packages (from torchvision) (1.21.2)
Requirement already satisfied: pillow!=8.3.0,>=5.3.0 in d:\ruanjian\python\lib\site-packages (from torchvision) (8.3.2)
Installing collected packages: typing-extensions, torch, torchvision, torchaudio
Successfully installed torch-1.10.1 torchaudio-0.10.1 torchvision-0.11.2 typing-extensions-4.0.1
PS C:\WINDOWS\system32>
```

(Optional) If your computer device is equipped with a graphics card, you can choose to intall CUDA first, and then install PyTorch under CUDA

To install CUDA

- For Windows/Linux, see https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64
- For macOS, see https://developer.nvidia.com/nvidia-cuda-toolkit-11_6_0-developer-tools-mac-hosts

The command to install PyTorch under CUDA 10.2 is shown below.



3. Install JupyterLab

- Please see the website to install JupyterLab (Pip is recommended for the installation) https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html

Run this command **pip install jupyterlab** in Windows PowerShell for Windows System.

```
PS C:\WINDOWS\system32> pip install jupyterlab
Collecting jupyterlab
  Downloading jupyterlab-3.2.8-py3-none-any.whl (8.5 MB)
     |████████████████████████████████| 8.5 MB 2.2 MB/s
Collecting packaging
  Using cached packaging-21.3-py3-none-any.whl (40 kB)
Collecting jupyter-server~=1.4
  Downloading jupyter_server-1.13.3-py3-none-any.whl (395 kB)
     |████████████████████████████████| 395 kB 6.4 MB/s
Collecting jinja2>=2.1
  Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 3.2 MB/s
Collecting tornado>=6.1.0
  Downloading tornado-6.1-cp39-cp39-win_amd64.whl (422 kB)
     |████████████████████████████████| 422 kB 6.4 MB/s
```

4. **Install required Python modules and packages**
   - Pip is recommended. The general command to install Python packages is 'pip install <package name>', such as (Please run the commands in Windows PowerShell for Windows System)
     - pip install numpy

```
PS C:\WINDOWS\system32> pip install numpy
Requirement already satisfied: numpy in d:\ruanjian\python\lib\site-packages (1
.21.2)
```

     - pip install matplotlib

```
PS C:\WINDOWS\system32> pip install matplotlib
Requirement already satisfied: matplotlib in d:\ruanjian\python\lib\site-packag
es (3.4.3)
Requirement already satisfied: numpy>=1.16 in d:\ruanjian\python\lib\site-packa
ges (from matplotlib) (1.21.2)
Requirement already satisfied: pillow>=6.2.0 in d:\ruanjian\python\lib\site-pac
kages (from matplotlib) (8.3.2)
Requirement already satisfied: python-dateutil>=2.7 in d:\ruanjian\python\lib\s
ite-packages (from matplotlib) (2.8.2)
```

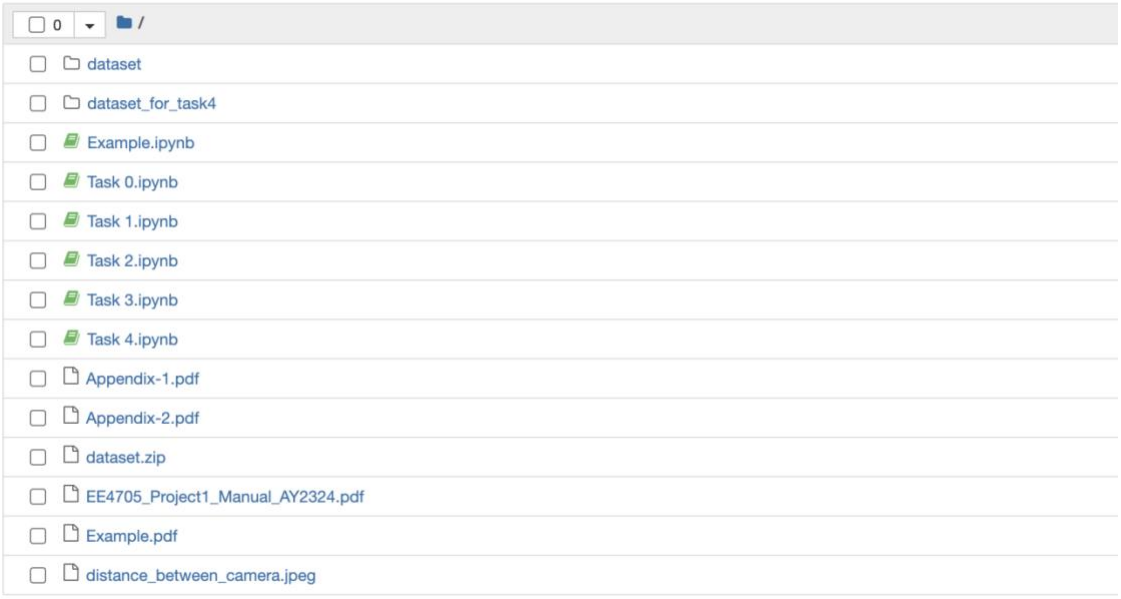
     - pip install opencv-python

```
PS C:\WINDOWS\system32> pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.5.5.62-cp36-abi3-win_amd64.whl (35.4 MB)
     |████████████████████████████████| 35.4 MB 6.8 MB/s
Requirement already satisfied: numpy>=1.17.3 in d:\ruanjian\python\lib\site-pac
kages (from opencv-python) (1.21.2)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.5.62
```

VII. Run JupyterLab to find our detailed materials for this project.

An instruction to run JupyterLab is given for Windows System. For the other systems, please google it how to run jupyterlab by yourself.

1. Enter ***jupyter-lab <project-files-local-absolute-path>***. For instance, enter ' jupyter-lab D:\EE4705_Project_1'. Please note that the path can not contain any spaces.
2. Then you will see a online jupyter workspace in your default browser. Please note that do NOT shut down the Windows PowerShell when doing your project.



3. Now you can see our detailed materials for this project, which includes a example and five tasks. What you need to do is to read the example first carefully and then finish the requirements in the five tasks.

VIII. References

Some useful manuals and documents are listed below for your ease reference, understanding necessary domain of knowledge and finishing the project smoothly.

1. For Python, https://docs.python.org/3/tutorial/
2. For PyTorch, https://pytorch.org/tutorials/
3. For JupyterLab, https://jupyterlab.readthedocs.io/en/stable/
4. **For building the neural network in PyTorch (Important. You probably need them)**
   a) https://pytorch.org/tutorials/beginner/basics/buildmodel_tutorial.html
   b) https://www.youtube.com/watch?v=XswEBzNgIYc
   c) https://towardsdatascience.com/pytorch-basics-how-to-train-your-neural-net-intro-to-cnn-26a14c2ea29
   d) https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/

IX. Appendix

Please see files *Appendix-1.pdf* and *Appendix-2.pdf*