EEE4490 Computer Vision and Image Processing

Lab 3: Deep Learning Models and its Applications

**Objective**: To learn techniques in image processing and computer vision using Python

**Tools**: Windows PC

**Software**:

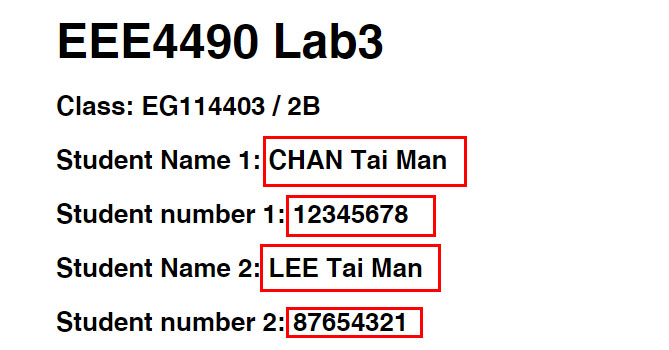
Anaconda

**Topics covered**:

* Perform pre-trained deep learning model in OpenCV
* Propose a real-world application using pre-trained deep learning model

Submission

1. Do your programming using the provide Jupyter Notebook file “Lab3\_Template.ipynb”. Remember to write your name on the top of the Jupyter Notebook. Produce a PDF file of the Jupyter Notebook. The name of the Jupyter should be “**Lab3\_[ChanTaiMan\_LeeTaiMan].pdf**”.



2. Write your proposal using the provided template file “Lab3\_Proposal01.docx”. Produce a PDF file of the proposal. The name of the PDF file should be “**Proposal\_[ChanTaiMan\_LeeTaiMan].pdf**”. Remember to write your name at the top of the proposal.

3. **Submit the two PDF files and the Jupyter Notebook to the Moodle by 4 Dec 2023 23:00**.

## Introduction

In this laboratory exercise, you are going to learn pre-trained deep learning model in OpenCV. This includes

1. object detection
2. face detection
3. face recognition
4. face expression recognition
5. human pose estimation
6. hand pose estimation
7. text detection
8. text recognition

Also, propose a real-world application using one or more pre-trained deep learning model(s).

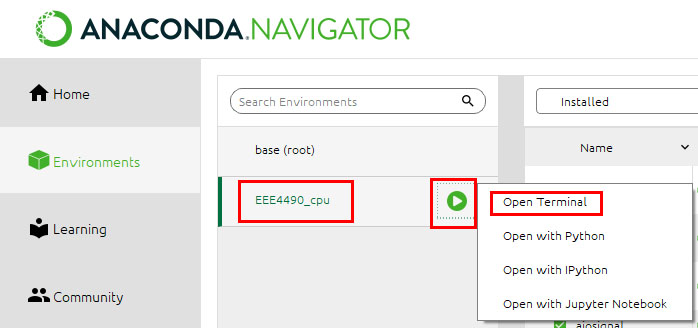
Two students form a group and each group submits a Jupyter notebook and a report.

You need to simplify the command to run the program in following section 2.1-3.4. For example, we do not want “**python demo.py --input imageA.jpg --model MODEL.onnx**”. Instead, we want “**python demo.py**”.

## 2.1. Object Detection (3 marks)

Modify the **demo.py** in “2\_1\_Object\_Detection” folder that can run the program to process one **image and** all the images captured by a **USB camera** without input argument.

In the anaconda, change to the “**EEE4490\_cpu**” environment. Click the green button and click “**Open Terminal**” as shown below.



Type the following command to upgrade the opencv-python

**pip install --upgrade opencv-python**

This will download and upgrade the opencv-python package. This needs to be done one time only.



Now, we run the demo.py program in the folder “2\_1\_Object\_Detection\data”. We go to the folder “[your path]\ToStudent\2\_1\_Object\_Detection” by the following command.

**cd [your path]\ToStudent\2\_1\_Object\_Detection**

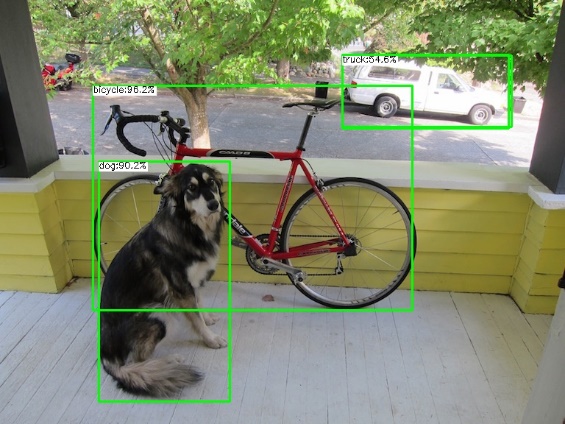
For example, in my case, I type command

**cd C:\Users\user\Downloads\ToStudent\2\_1\_Object\_Detection**

Then, type the following command for processing one image named testing\_image.jpg stored in “2\_1\_Object\_Detection\data” folder.

|  |
| --- |
| **python demo.py --input data/testing\_image.jpg --model object\_detection\_yolox\_2022nov.onnx --save** |

This command will give you an output image “result.jpg” stored in the folder “2\_1\_Object\_Detection”.



To run the program processing all the images captured by a USB camera, we can run the following command. Hit any key on the keyboard to stop the program.

|  |
| --- |
| **python demo.py** |

*This command may take 1 minutes to start working. Please wait*.

Now, we modify the code in demo.py and put them in the “Lab3\_Template.ipynb” notebook file. This modification should allow user to run the program without using command arguments. See the Jupyter notebook “Lab3\_Template.ipynb” session 2.1 as an example. You do the following questions in a similar manner in Jupyter Notebook.

## 2.2. Face Detection (3 marks)

Modify the **demo.py** in “2\_2\_Face\_Detection” folder that can run the program in **image and** **USB camera** **source** without input argument.

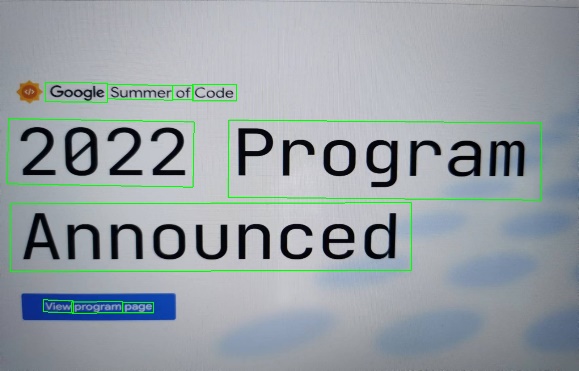


## 2.3. Person Detection (3 marks)

## 一張含有 體育, 文字, 平衡, 體適能 的圖片 自動產生的描述

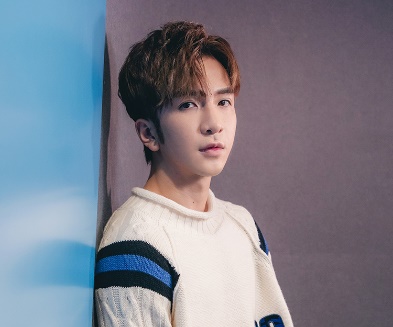
## 2.4. Text Detection (3 marks)

Modify the **demo.py** in “2\_4\_Text\_Detection” folder that can run the program in **image and** **USB camera** **source** without input argument.



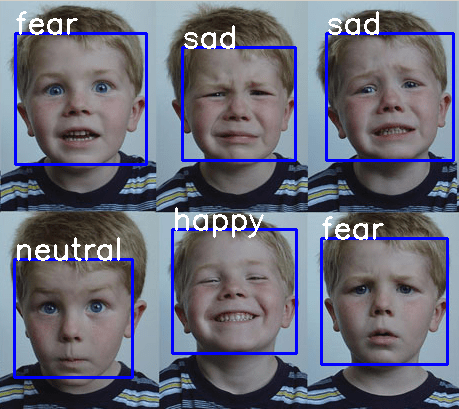
## 3.1. Face Recognition (4 marks)

Modify the **demo.py** in “3\_1\_Face\_Recognition” folder that can run the program in **a pair of images** without input argument.

## 3.2. Face Expression Recognition (6 marks)

Modify the **demo.py** in “3\_2\_Face\_Expression\_Recognition” folder that can run the program in **image and** **USB camera** without input argument.



## 3.3. Hand Pose Estimation (6 marks)

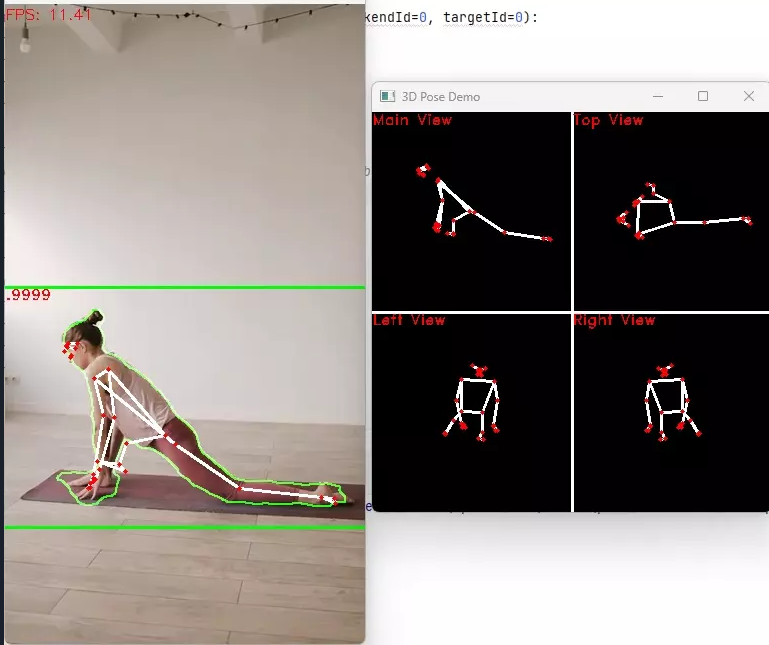
Modify the **demo.py** in “3\_4\_Hand\_Pose\_Estimation” folder that can run the program in **image and** **USB camera** **source** without input argument.

一張含有 螢幕擷取畫面, 人的臉孔, 人員, 設計 的圖片

自動產生的描述

## 3.4. Human Pose Estimation (6 marks)

Modify the **demo.py** in “3\_4\_Human\_Pose\_Estimation” folder that can run the program in **image and** **USB camera** **source** without input argument.



## 3.5. Text Recognition (6 marks)

Modify the **demo.py** in “3\_5\_Text\_Recognition” folder that can run the program in **image and** **USB camera** **source** without input argument.



## 4. Propose a real-world application using pre-trained deep learning model or/and image processing technologies (60 marks)

Please include the following aspects.

1. A story of real-world application. Detail the problem to be solved. (10 marks)
2. A block diagram shows the method to solve the problem. (10 marks)
3. Description of using the technologies to solve the problem. (20 marks)
4. Required hardware or setup which can facilitate/improve the performance of the system. (20 marks)