**EIE4105 Multimodal Human Computer Interaction Technologies**

**Lab: Harness AI with Nvidia Jetson TX2**

**A. Objectives and Outcomes**

After finishing this lab, you should be able to perform the following

* Use the Nvidia Jetson TX2 Developer Kit, TensorRT, Jetson Inference Library, and OpenCV to implement real-time image classifiers.
* Use Tensorflow and Keras to develop deep neural networks and convolutional neural networks for handwritten digit classification.
* Deploy the DNN and CNN for handwritten digit classification on Jetson TX2 and perform classification in real-time.

**B. Assessment Criteria**

* Able to use and develop DNN and CNN on the Jetson TX2.
* Able to demonstrate a real-time image classifier.
* Able to explain the capability of different deep neural networks.
* Able to write a clear report.

**C. Submission and Demonstration**

* Copy and paste the graphs and images that you obtain in this lab and **convert it to PDF**.
* Submit your report to Blackboard before the deadline specified in Blackboard.
* Demonstrate your real-time classifiers to the lab instructors.

**D. Jetson TX2 Developer Kit**

The Jetson TX2 is a single-board computer equipped with an NVIDIA Pascal and two ARM v8 64 bit CPUs, 8G DDR4 RAM, and 32G flash storage. You may deploy pre-trained deep learning models on the board and use it as an inference engine. You may also use it to train deep learning models. For detailed specification, visit https://developer.nvidia.com/embedded/jetson-tx2-developer-kit.

**E. Procedures**

***E.1 Use OpenCV to Display Live Video***

1. Switch on the Jetson TX2 by pressing the right-most button shown below.



1. When you see the login screen, log in the system using "student" and "student" as the username and password, respectively.
2. After you see the desktop of Ubuntu Linux, clock the icon  on the top-bar. Then, select “More Networks”, then select “Wi-Fi.HK via PolyU” (this connection is slow) or the SSID as instructed by the lab instructor.
3. Click the top-left icon ****and search for “Terminal." Select Terminal to open a shell terminal and ensure that you are in your home directory (use the pwd command to check or use the cd command to return to your home directory.

****

1. Start the “Chromium Web Browser” . Then, visit  
   <http://www.eie.polyu.edu.hk/~mwmak/notes/EIE4105/jetson-lab.tgz> using “student” and “learning” as the username and password, respectively.
2. Decompress the .tgz file to your home folder as follows:[[1]](#footnote-1)

$ cd ~

$ mv ~/Download/jetson-lab.tgz .

$ tar zxvf jetson-lab.tgz

You should see the folder ~/jetson-lab under your home folder.

1. Change directory to openCV and start the Spyder Integrated Development Environment (IDE):

$ cd ~/jetson-lab/opencv

$ spyder &

1. In Spyder, open the file “webcam.py” under the folder “opencv”. This script captures live video from the USB webcam, converts the video to gray level, and displays the live video on a window. Study the Python code as you may need it in the latter part of the lab. Run the script by typing the following command in the Terminal:

$ python3 webcam.py

Type “q” to quite the program.

1. Open the file “CannyDetection.py." This script calls the Canny edge detector of OpenCV and displays the edge of live video. Execute the script in the Terminal as follows:

$ python3 CannyDetection.py

Press the “Esc” key to quite the program.

***E.2 Image Classification using Pre-Trained CNN***

1. Change the folder to “imagenet” and open the file “imagenet-console.py”. This script reads a .jpg file and determines the object in the file. Execute the script as follows:

$ cd ~/jetson-lab/imagenet

$ python3 imagenet-console.py images/polar\_bear.jpg

$ python3 imagenet-console.py images/brown\_bear.jpg

$ python3 imagenet-console.py images/black\_bear.jpg

You shall see the recognition results in the Terminal. To display the image, type

$ xdg-open images/polar\_bear.jpg

Put the images into your report and state the recognition results (including the class name and confidence level).

1. Download some more images (.jpg) files from the ImageNet ([www.image-net.org](http://www.image-net.org)) and see if the program can recognize the objects. Put the images into your report and state the recognition results (including the class name and confidence level). You may also find some images in <http://www.eie.polyu.edu.hk/~mwmak/notes/EIE4105/images.zip>
2. Open the file “imagenet-camera.py” under the folder “imagenet”. This script can recognize objects in live video. Point the webcam to objects such as a mouse, keyboard, monitor, mobile phone to see if it works. Study the code carefully to see how to use the jetson-inference library to load a pre-trained network to perform real-time classification of objects in live video. Execute the program by typing

$ python3 imagenet-camera.py --camera /dev/video1

Press “Ctrl-C” in the Terminal to quite the program.

***E.3 DNN and CNN for Handwritten Digit Recognition***

1. Change directory to mnist by typing

$ cd ~/jetson-lab/mnist

1. Open the file “mnist\_dnn.py”. This script trains and tests a deep neural network (DNN) using Keras and Tensorflow. Read the code carefully to understand how to implement a DNN. Execute the script as follows:

$ python3 mnist\_dnn.py

The program will automatically download the MNIST dataset from Amazon and store it in “~/.keras/datasets/”. If your internet connection is very slow, it will take a while to download the file (11Mbytes). You may download the dataset from the EIE server as follows:

$ cd ~/.keras/datasets

$ wget --user student --password learning \

http://www.eie.polyu.edu.hk/~mwmak/notes/EIE4105/mnist.npz

Change the number of hidden layers and the number of nodes in the hidden layers to see if the changes could improve the classification accuracy. Report the accuracy you obtain.[[2]](#footnote-2)

1. Open the file “mnist\_cnn.py”. This script trains and tests a convolutional neural network (CNN). Read the code carefully to understand how to implement a CNN. Execute the script as follows:

$ python3 mnist\_cnn.py

The script also saves the CNN in .h5 format in the folder “mnist/models”. Change the number of layers and the number of nodes to investigate how these parameters affect performance. Also, change the activation function to see if you can get better performance.

1. Open the file "mnist\_console.py." Study the code carefully to see how to use a pre-trained CNN for handwritten digit recognition from .jpg files. Execute the program as follows:

$ python3 mnist\_console.py images/test/0/1001.jpg

$ python3 mnist\_console.py images/test/2/5921.jpg

1. Open the file "mnist\_camera.py." Use "opencv/webcam.py” and “mnist/mnist\_console.py" as examples to implement a real-time handwritten digit classifier. You may display some digits on-screen by reading the file "mnist/images/mnist-digits-images.pdf.”

$ cd ~/jetson-lab/mnist

$ xdg-open images/mnist-digits-images.pdf

Click the icon  in xdg-open so that the pages can fit into the window. You may point your webcam to the image to see if your “mnist\_camera.py” works. Report your observations by capturing the screen.

References:

[1] <https://github.com/dusty-nv/jetson-inference#hello-ai-world>

-- END --

1. In this lab, the $ symbol means the shell prompt of a terminal. [↑](#footnote-ref-1)
2. Training a CNN on Jetson is slow. You do not need to try too many combinations. [↑](#footnote-ref-2)