You are free to select either of the following two projects as your AI6121 course project.

Gender and Age Detection

Gender and age detection aims to guess the gender and age from a single image (face image) of a person. The predicted gender will be binary, i.e. either ‘Male’ or ‘Female’. The age prediction can be formulated as a classification problem by classifying ages into one of the following ranges: (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), and (60 – 100). Alternatively, you can also formulate it as a regression problem by predicting the exact age given a face image. The regression approach is more challenging due to the limited information (a single face image) and various noises in facial images. It is up to you to put it as a classification or regression task.

You can adopt either multi-layer perceptron (MLP) or convolutional neural networks with different network backbone to develop your gender and age detection system. You are free to use different programming languages, software libraries and functions, etc. in your implementation. The dataset is available at: https://www.kaggle.com/ttungl/adience-benchmark-gender-and-age-classification or other websites.

This project consists of the following tasks:

1) Design and develop your gender and age detection network. Train your network over the training set, evaluate over the test set, and report your evaluation results. Your project report should include detailed description and discussion of your network design as well as your network evaluation. For the source codes, you can either append them at the end of the project report, or submit them separately.
2) Investigate different hyper-parameters and design options such as learning rate, optimizers, loss functions, etc. to study how they affect the network performance. The report should include detailed description and discussion of your study.
3) Benchmark with the state-of-the-art, and discuss the constraint of your designed network and possible improvements. Implement and verify your ideas over the test set. This subtask is **optional**, and there will be bonus marks for nice addressing of this subtask.

You need to submit your project report in PDF format, and there is no standard report template. Note you can refer to some publicly available resources like open source codes, but cannot copy those open source materials as your implementation. Please include relevant reference if you adopt certain codes from various public sources otherwise it forms plagiarism.

Please submit your solution through NTULearn before the deadline on 20 Nov 2020. There will be penalty for late submissions.

MNIST handwritten digit recognition

The handwritten digit recognition is the ability of computers to recognize human handwritten digits. This project will work on the dataset MNIST, which is probably one of the most popular datasets among machine learning and deep learning enthusiasts. The MNIST dataset contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing. It thus has 10 different classes, where each grayscale digit image is represented by a 28×28 matrix. You can download the dataset from the website: https://www.kaggle.com/sachinpatel21/az-handwritten-alphabets-in-csv-format or other websites.

You can adopt either multi-layer perceptron (MLP) or convolutional neural networks with different network backbone to develop your handwritten digit recognition system. You are free to use different programming languages, software libraries and functions, etc. in your implementation.

This project consists of the following tasks:

1) Design and develop your handwritten digit recognition network. Train your network over the MNIST training set, evaluate over the test set, and report your evaluation results. Your project report should include detailed description and discussion of your network design as well as your evaluation results. For the source codes, you can either append them at the end of the project report, or submit them separately.
2) Investigate different hyper-parameters and design options such as learning rate, optimizers, loss functions, etc. to study how they affect the network performance. The report should include detailed description and discussion of your study.
3) Benchmark with the state-of-the-art, and discuss the constraint of your designed network and possible improvements. Implement and verify your ideas over the test set. This subtask is **optional**, and there will be bonus marks for nice addressing of this subtask.

You need to submit your project report in PDF format, and there is no standard report template. Note you can refer to some publicly available resources like open source codes, but cannot copy those open source materials as your implementation. Please include relevant reference if you adopt certain codes from various public sources otherwise it forms plagiarism.

Please submit your solution through NTULearn before the deadline on 20 Nov 2021. There will be penalty for late submissions.