**Face Verification - 2017 Summer**

1. **What’s the task**

You need to design an algorithm to determine whether two face images come from the same person. We will use an unopened dataset to assess your algorithm.

1. **Dataset**

In the “webface” folder. “webface” is an open dataset, containing 10k+ people’s face images. How many instances to use is up to you, and you are allowed to use extra data if you find some. The folder name can be viewed as person ID, so we don’t provide label file, which you can make by yourself quite easily.

Note:

* The faces in dataset have been aligned.
* If you use extra data, we provide an existing face detection and alignment algorithm, named MTCNN. See usage below.

1. **What to hand in**
2. Code with necessary libs or other stuff to make it work. Please keep the interface protocol:

*Y = FaceVerification(X1, X2)*

X1 and X2 are the path of two face images to verify,

Y∈{1, 0}, where 1 means X1 and X2 come from the same person, 0 means not.

1. A brief and clear experiment report, including but not limited to

* How to run your code, a short example will be appreciated.
* Your method and its novelty.

1. Please turn in your work into folder ‘UpLoad’, with the form of .rar file , named like this:

‘组长学号-组员1学号-组员2学号.rar’

We will use a script to detect who has turned in this homework, so any violation of this name rule may result in ZERO point of this homework. Just be careful with this.

1. **Notes**
2. You are allowed to use ANY method to solve the problem, including different deep learning platforms.
3. DDL：2017/12/15
4. Methods based on deep learning probably outperform traditional machine learning methods, but since deep learning heavily depends on your GPU, where different team may have different computing resources, it seems unfair to judge your grade based on the performance of your algorithm only. So we also value your exploration or fantastic ideas about the problem, which should be stated clearly in your experiment report.
5. Your grade = 60% \* score\_of\_your\_report + 40% \* score\_of\_your\_verification\_accuracy, so be serious with your report.
6. Name your report with a standard name among “report.docx, report.doc, report.txt, report.md”, do not use Chinese words. In your report, remember to give your team member names and student IDs.
7. Honor Code: do NOT just download an existing face verification model (.caffemodel or something like that) and then turn it in. But you are allowed to fine-tune the model, and if you do fine-tune some model, please mention it in your experiment report. ANY cheating will lead to zero point of this assignment.
8. **MTCNN Usage**

MTCNN algorithm we provide is from an open program in “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks” (To learn more, please refer to the paper).

**Steps:**

1. You need to install caffe and compile matlab interface firstly. Please refer to caffe official website.
2. Run the ‘*FaceAlignment\_MTCNN/code/codes/MTCNNv2/demo2.m*’. You only need to modify some basic path setting in ‘*demo2.m*’, like:

*caffe\_path='~/Caffe/Caffe\_default/matlab';*

*input\_dir = '~/Caffe/MTCNN\_FaceAlignment/webface\_samples'*

*output\_dir = '~/Caffe/MTCNN\_FaceAlignment/webface\_samples\_aligned'*

If you don’t use GPU, use *caffe.set\_mode\_cpu().* Then you should be able to run. Enjoy!