

Automating Roll Calls with Deep Metric Learning

Guan-Zhong Wang (U10516045)

Supervisor: Chun-Ming Tsai

January 14, 2019

Abstract

In my semester project, I prototyped a roll call system called **pyRollCall** in Python 3.5 to automate the traditional roll call procedures with deep metric learning.

Normally, a teacher needs to call each student's name one by one in order to check who is present at a class. This system aims to simplify the traditional roll call procedures by pre-computing each student's facial embeddings using **dlib**, so that it can identify students' faces correctly later with **OpenCV** as well as **face_recognition** module and completely automates roll calls.

1 Introduction

With the rapid growth of technology, more and more repetitive tasks can be automated by programs. In my semester project, I built a system to replace the tedious procedure of a traditional roll call.

Initially, the users (typically teachers) are required to populate the database with the information of the courses and students he/her teaches. Secondly, the users have to take several photos of each student presented in the database and compute their facial embeddings. Lastly, the program can start identifying students' faces and help users automate roll calls.

2 Implementation

This system employs a technique called **deep metric learning** (i.e., deep-learning-based facial recognition) to quantify the faces of students via dlib and face_recognition module. In order to achieve facial recognition in Python and OpenCV, I referenced the tutorial [1] on pyimagesearch by Adrian Rosebrock.

According to the tutorial [1], the network architecture for face recognition being used in this system is based on ResNet-34 from the *Deep Residual Learning for Image Recognition* paper [2] by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. The network in dlib was already trained by the creator of dlib, Davis King, on a dataset of approximately three million images, while face_recognition is a wrapper of dlib which enables users to use it easily.

Users can manage their data, including courses, students and facial embeddings through GUI. They can also initiate a roll call through GUI, letting students sign in to a class with facial recognition, as shown in Figure 1.

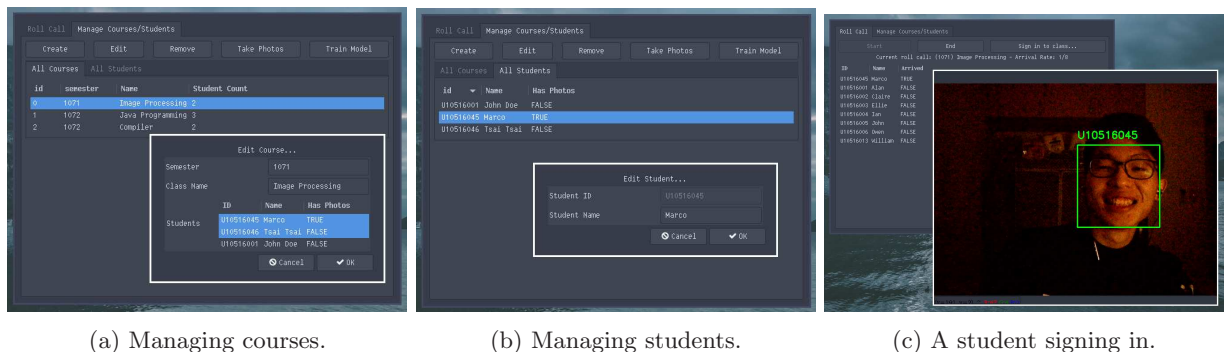


Figure 1: pyRollCall's GUI, database and facial recognition.

3 Experimental Results

I experimented this system on 7 people, including 6 famous actors from the movie Jurassic Park (1993) and myself, with each of them given a student ID (e.g., U10516001). The training image set of the 6 actors is from the tutorial [1] on pyimagesearch.

1. U10516001 - Alan Grant (22 images)
2. U10516002 - Claire Dearing (53 images)
3. U10516003 - Ellie Sattler (31 images)
4. U10516004 - Ian Malcolm (41 images)
5. U10516005 - John Hammond (36 images)
6. U10516006 - Owen Grady (35 images)
7. U10516045 - Guan-Zhong Wang (5 images)

It can distinguish between 7 different people, successfully automating a roll call, as shown in Figure 2.

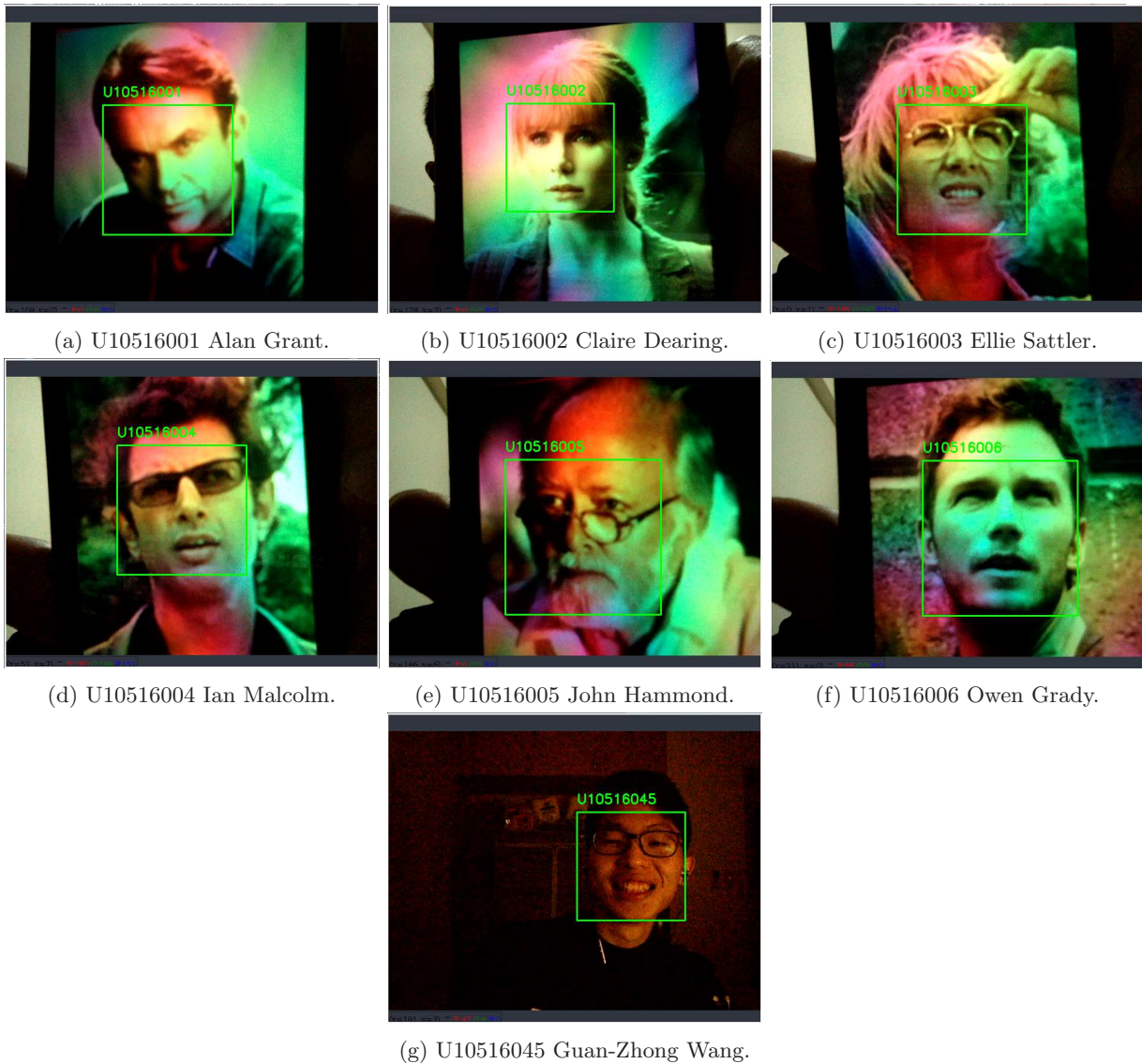


Figure 2: Experimental Result.

4 Discussion

Since this system is merely a prototype, it still leaves a lot to be desired. For example, I have not confirmed how many photos must be provided per person in order to make the system recognize each person correctly. Too few photos will lead to false facial recognition.

There might also be some alternative approaches that can improve the accuracy of facial recognition. These issue can be carried on in subsequent studies.

5 References

- [1] tutorial on pyimagesearch.com (<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>) by Adrian Rosebrock.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition", 2015 (<https://arxiv.org/abs/1512.03385>)