Facial Recognition Attendance Checker

Thanet Prompinit, Salisa Cheawcharnthong, Pornchai Mongkolnam and Jonathan H. Chan
School of Information Technology
King Mongkut's University of Technology Thonburi
Bangkok 10140, Thailand
thanet.prom@mail.kmutt.ac.th, salisa_cct@hotmail.com, pornchai@sit.kmutt.ac.th, jonathan@sit.kmutt.ac.th

***Abstract*— This work presents a novel facial recognition framework for attendance checking, by using the student's smartphone and Bluetooth-low-energy (BLE) beacons placed in the classroom. Facial recognition is used for authentication based on the Active Appearance Model (68-point facial landmarks) that is converted to a 128-dimensional vector space. One or more beacons are used to track the sitting position of each student. Then image processing is used to analyze the current facial sentiment of the student. Facial recognition models based on Naïve Bayes, Support Vector Machine, and Random Forest were compared. The best model was found to be Random Forest with an accuracy of about 98% on the test data. A web-based application has been deployed and tested on Android smartphones to connect with the API for the needed services.**

I. Introduction

According to a recent (2016) statistical survey, only about 3 percent of the Thai population, or just fewer than 2 million students, pursued higher education [1,2]. However, the global approaches to higher education learning have been evolving rapidly with advances in information technology, leveraging the use of videos, graphics and animation, text messages, as well as social networking applications. For example, the traditional e-learning or distance-based education has evolved in form of a more comprehensive Massive Open Online Course (MOOC); such offering is available for anyone with Internet access for free, with commercial entities offering certification for a small fee [3]. Even with the rising popularity of MOOCs, online learning in general lacks the physical interactive element found in a traditional classroom. Currently, in most Thai universities, attendance is still mandatory. Nonetheless, one can argue that it is a burden for the instructors to manually keep students' attendance, even with the use of attendance checking devices such as fingerprint reader or radio frequency identification (RFID) card reader. Also, students still do not have timely access to their attendance record. In order to address the above issues, an application was developed as a proof of concept based on Android smartphone, beacon(s) and a rudimentary facial recognition procedure for efficient classroom attendance checking.

The aim of this work is to assist both students and instructors to seamlessly keep attendance and participation records to improve and facilitate the classroom learning environment, with the added bonus of reducing the instructors' burden in keeping the attendance. One or more beacons with Bluetooth Low Energy (BLE) are used to triangulate each student's position during the class. The instructor can also initiate an authentication protocol at any given time, if he/she wishes, to make sure that a student is present in class. In addition, more subtle analyses such as facial sentiments, individual and group sitting positions, can be used to help educators understand their students' behaviors in classrooms better.

II. Methodology

Our proposed SMATCH: Smart Classroom Attendance Checker system comprises a web-based API facial recognition server, a Firebase server, and an Android smartphone with Internet access and connected to a beacon [4]. One or more beacons are placed inside the classroom for the purpose of attendance checking. At least one beacon is needed but the uncertainty can be up to one meter. Thus we have tested up to three beacons in order to improve precision of student position triangulation. For the purpose of analytics though, an approximate sitting position that depicts the front of the classroom, middle area, or back of the classroom may be adequate. Each student is required to register his or her phone and number with our system. This process could be initiated before a semester begins. When the student is coming to class and within the beacon signal coverage area, the attendance check-in and sitting position data are recorded to the online Firebase server. An instructor can perform a facial check-in at a specific time or at a random interval. For example, whenever there is a quiz or in-class exercise, students need to check their attendance with facial images taken by the smartphones. When initiated, the instructor's smartphone sends a push notification to the students' smartphones, which asks for a facial check-in in order to authenticate the genuinely enrolled students, not someone else. In facial recognition, our method is based on the Active Appearance Model (AAM) and some well-known data mining techniques for classifying students' faces. The system's facial recognition framework is shown in Fig. 1.

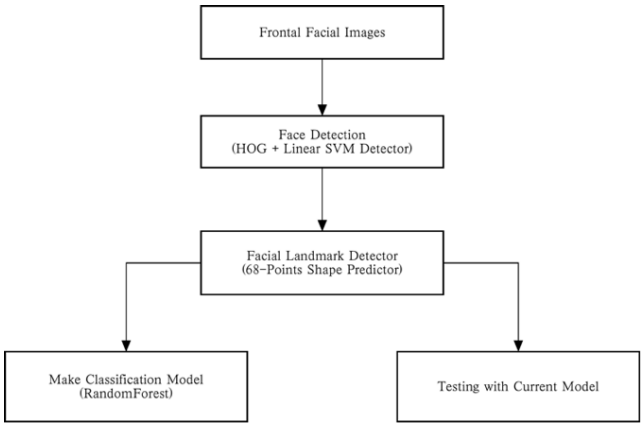

Fig. 1. Framework of basic facial recognition system

A more detailed description about the framework of the facial recognition system is provided as follows.

A. Frontal Facial Images

When students use the system for the first time, they must take 5 self-facial images for making a recognition model at the positions of center, right, left, up, and down. The next time the students would only need to use a single self-facial

978-616-368-065-5 ©2018 JSCI

image for comparison with the existing pre-trained model created earlier.

B. Face Detection

When the system gets an image from students. The system makes detection and crops just a facial area from whole image by using HOG + Linear SVM Detector techniques from the DLib Library.

C. Facial Landmark Detector

In this step, the system detects 68-point facial landmarks on the face image and then extracts these data for classification in the next step.

D. Classification Model

When the system receives facial landmark data for the first time, data mining techniques are used for making a model of classification of each student for authentication purpose. For the next time of usage, we would have unseen data for authentication purpose.

For the facial recognition process, Python was used to develop a web-based API server, with sending and receiving messages using the HTTP protocol, and the data were stored in the Java Script Object Notation (JSON) format. This process may be termed a RESTful web service [5]. The Python web framework known as Flask [6] was used. For the feature extraction task in the facial recognition framework, we used DLib [7] for both face detection and face landmark, and then we used the NumPy module [8] to convert the 68-point face landmarks to 128-dimensional data. The popular WEKA data mining software [9] was used for classification purpose. A connector between Python and WEKA was developed for the facial recognition process. Fig. 2 shows the schematic of the facial recognition system.

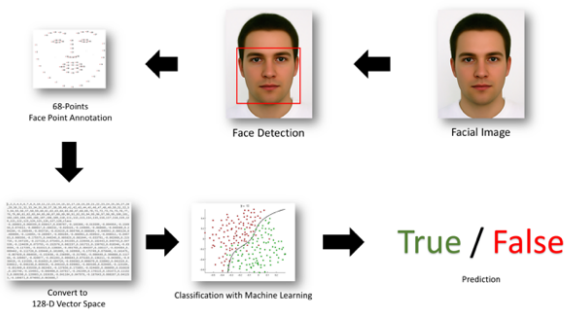

Fig. 2. Schematic of facial recognition system.

III. Results

In training a facial dataset, we applied the AAM technique used for feature extractions in order to obtain the 68-point face landmarks and converting them to 128 dimensions for subsequent classification [10,11]. For the dataset, we used 5 correct face samples and 5 incorrect face samples, for each of the 10 test subjects, and that resulted in a total of 100 sample faces. The samples were obtained from the publicly available extended Cohn-Kanade face database [12]. The dataset was trained with three common classifiers: Naïve Bayes, Support Vector Machine, and Random Forest. Table 1 shows comparisons of the three classifiers in terms of the accuracy of prediction. As can be seen from the results, Random Forest performed better than the other classifiers, for both training and testing data (shown in bold). The testing dataset was randomly generated from the same face database (not part of those 100 sample faces), which included 5 correct facial images and 5 incorrect facial images for each person, resulting in another total of 100 faces used for model testing.

Classifier	Training	Testing
Naïve Bayes	99%	68%
Support Vector Machine	76%	96%
Random Forest	**100%**	**98%**

Table 1. A comparison of the accuracy of different classifiers

References

[1] Office of The Higher Education Commission, Thailand. (2016). *Total Student Information.* Retrieved April 6, 2017, from Office of The Higher Education Commission, Thailand: http://www.info.mua.go.th/information/download.php?file_id=201706301624.xlsx&stat_id=344&id_member=.

[2] The Bereau of Registration Administration (BORA), Department of Provincial Administration, Thailand. (2016). *Number of Population in The Kingdom, Following by The Evidence of Registration*. Retrieved July 20, 2017, from The Bereau of Registration Administration (BORA), Department of Provincial Administration, Thailand: http://stat.bora.dopa.go.th/stat/y_stat59.htm.

[3] Glušac, D., Karuović, D., Milanov, D., Gluac, D., Karuovi, D., & Milanov, D. (2015). Massive open online courses - Pedagogical overview. *Carpathian Control Conference (ICCC), 2015 16th International*, (vi), 142–146. https://doi.org/10.1109/CarpathianCC.2015.7145063.

[4] Prompinit, T., Mongkolnam, P., Cheawcharnthong, S., & Chan, J. H. (2018). SMATCH: Smart Classroom Attendance Checker Using Beacon, Smartphone and Data Mining Techniques. *International Conference on Learning Innovation in Science and Technology (ICLIST 2018)*, March 21-24, 2018, Hua Hin, Thailand, pp. 1-8.

[5] Richardson, L., & Ruby, S. (2008). *RESTful Web Services*. https://doi.org/0596554605.

[6] Lewandowski, C. M. (2015). *Flask Web Development. The effects of brief mindfulness intervention on acute pain experience: An examination of individual difference* (Vol. 1). https://doi.org/10.1017/CBO9781107415324.004.

[7] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758. https://doi.org/10.1145/1577069.1755843.

[8] Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37.

[9] Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., & Trigg, L. (2005). WEKA: A Machine Learning Workbench for Data Mining. *Springer US. In: Data Mining and Knowledge Discovery Handbook*, 1305–14. https://doi.org/10.1007/0-387-25465-X_62.

[10] Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 681–685. https://doi.org/10.1007/BFb0054760.

[11] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Review of FaceNet : A Unified Embedding for Face Recognition and Clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–2. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Schroff_FaceNet_A_Unified_2015_CVPR_paper.pdf (Hess, 2017).

[12] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The extended Cohn-Kanade dataset (CK+): A complete facial expression dataset for action unit and emotion specified expression. *Cvprw*, (July), 94–101. https://doi.org/10.1109/ISIEA.2010.567950