Advanced Project for AI Convergence

Combining Video Class Solution with Face Recognition Solution -- ㈜케이테크

Huynh Van **Thong** – 207081

Final Report

Demonstration video: https://1drv.ms/v/s!AoeAp4aV23Gqeq-5PvynHUotpn8?e=TegyU2
Source code: https://github.com/th2l/FacialAnalysis-GUI

Project description

Combining Video Class Solution with Face Recognition Solution - ㈜케이테크

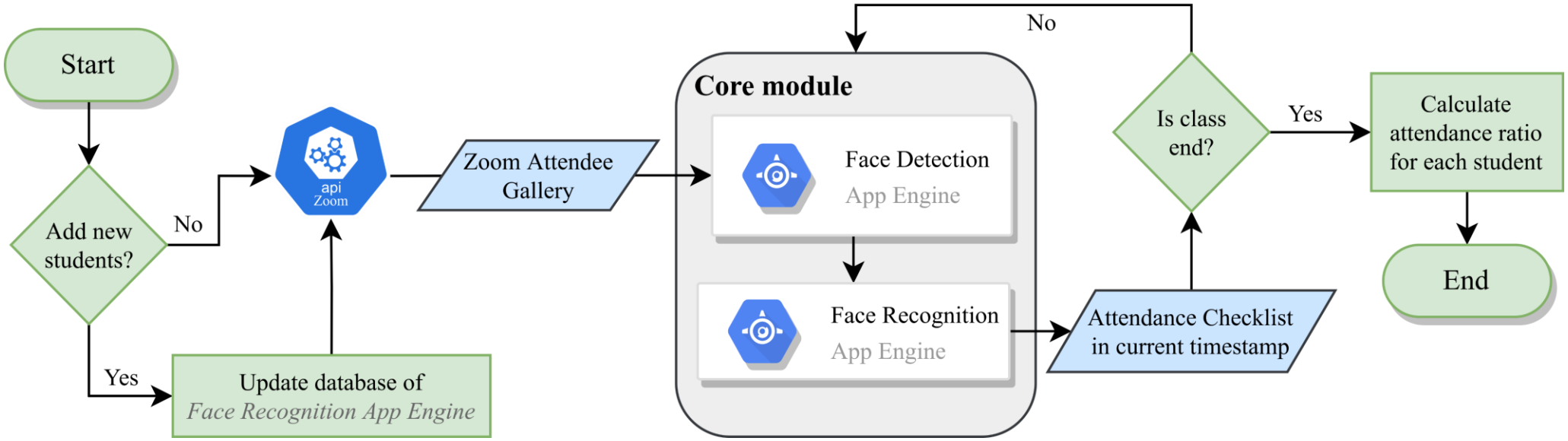
Task objective:

- Authenticate the class participation of students during online classroom (e.g., Zoom).

Input: recorded video from Zoom service, local video, webcam or other sources.

Output: Determine whether students are participating or not.

Face recognition overview



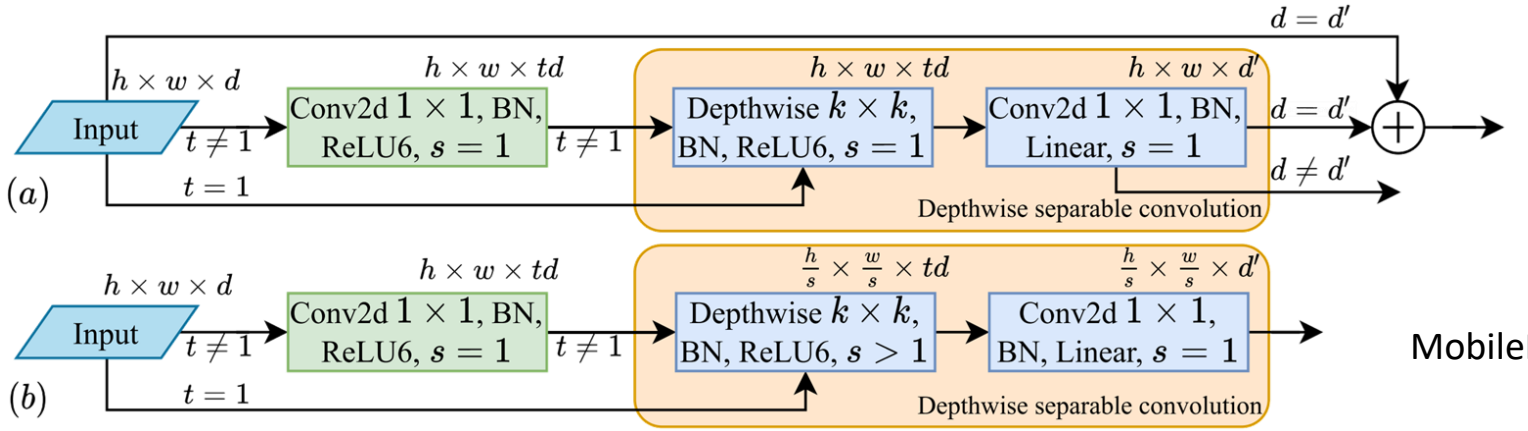
- ✓ **Face detection engine:** SCRFD [1].
- ✓ **Face recognition engine:** MobileNetV2 [2]**.**

[1] Guo, Jia, Jiankang Deng, Alexandros Lattas, and Stefanos Zafeiriou. "Sample and Computation Redistribution for Efficient Face Detection." arXiv preprint arXiv:2105.04714 (2021).

[2] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR, 2018.

Face recognition

- ✓ **Base model**: MobileNetV2 [1], number of parameters ~3.4M.
- ✓ **Training dataset**: MS1M-ArcFace dataset [2, 3, 4] (85k identities / 5.8M images).
- ✓ **Evaluation** with *TAR@FAR=1e-4* on 2 different dataset.
  - LFW: **0.996**
  - CFP_FP: **0.952**
  - AgeDB_30: **0.959**
  - IJB-B (IARPA Janus Benchmark-B Face Dataset): **0.887**
  - IJB-C (IARPA Janus Benchmark-B Face Dataset): **0.911**
- ✓ Implementation: TensorFlow + Keras.



MobileNetV2 building block [1]

[1] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR, 2018.

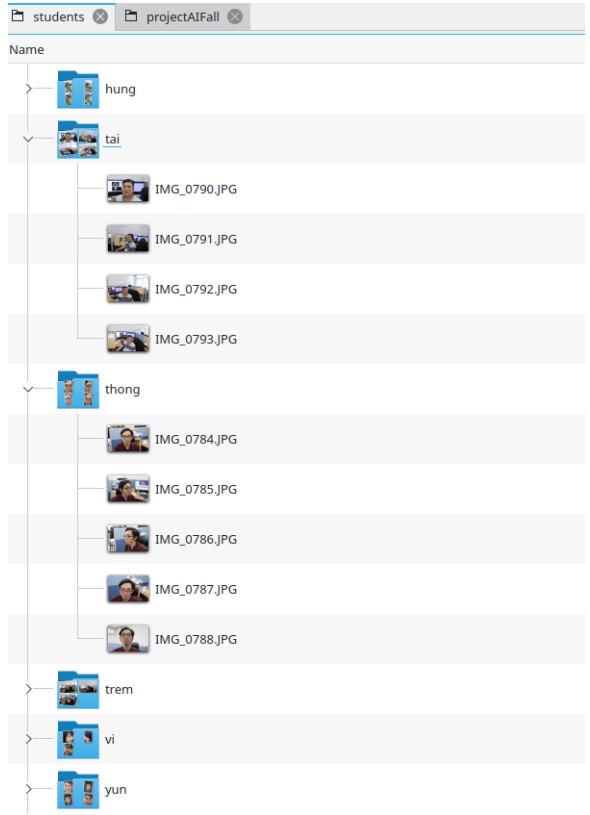
[2] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. ECCV, 2016.

[3] Jiankang Deng, Jia Guo, Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition, arXiv:1801.07698, 2018.

[4] https://drive.google.com/file/d/1SXS4-Am3bsKSK615qbYdbA_FMVh3sAvR/view

Face recognition on new users

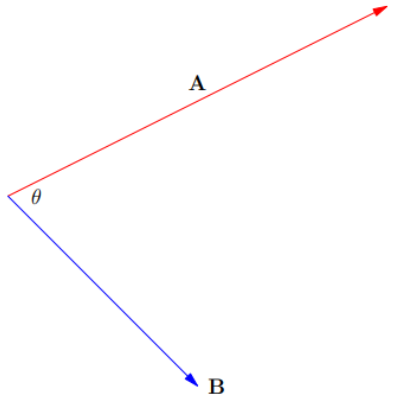
- Need: a folder contain face images of users (from different view of camera) which organizing in sub-folders.
- Process of the system:
  - ✓ **Step 1**: check user folders and calculate face embedding with pretrained MobileNetV2 (previous slide) and save to file.
  - ✓ **Step 2**: receive frame/image from webcam/video and do face detection, calculate face embedding.
  - ✓ **Step 3**: compare face embedding in step 2 with face embeddings in step 1, and find the one with minimum distance (e.g., L2).
  - ✓ **Step 4**: replace step 2 and 3 until no more frame/image.

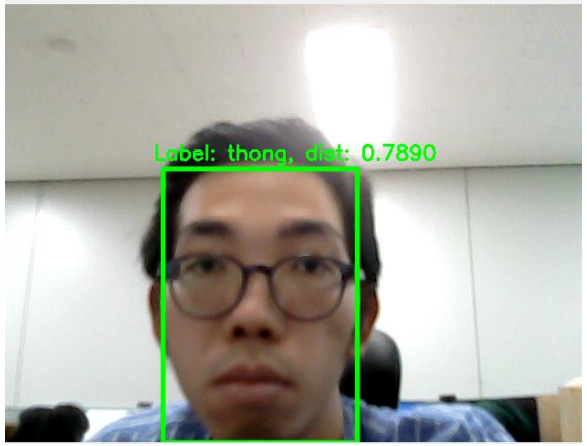


Face recognition on new users

- **Step 3**: *compare face embedding* in step 2 with face embeddings in step 1 and find the one with minimum distance.
  - Using machine learning (k-mean clustering, SVM, …)
  - Using simple distance: L2, ***cosine similarity***.

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$



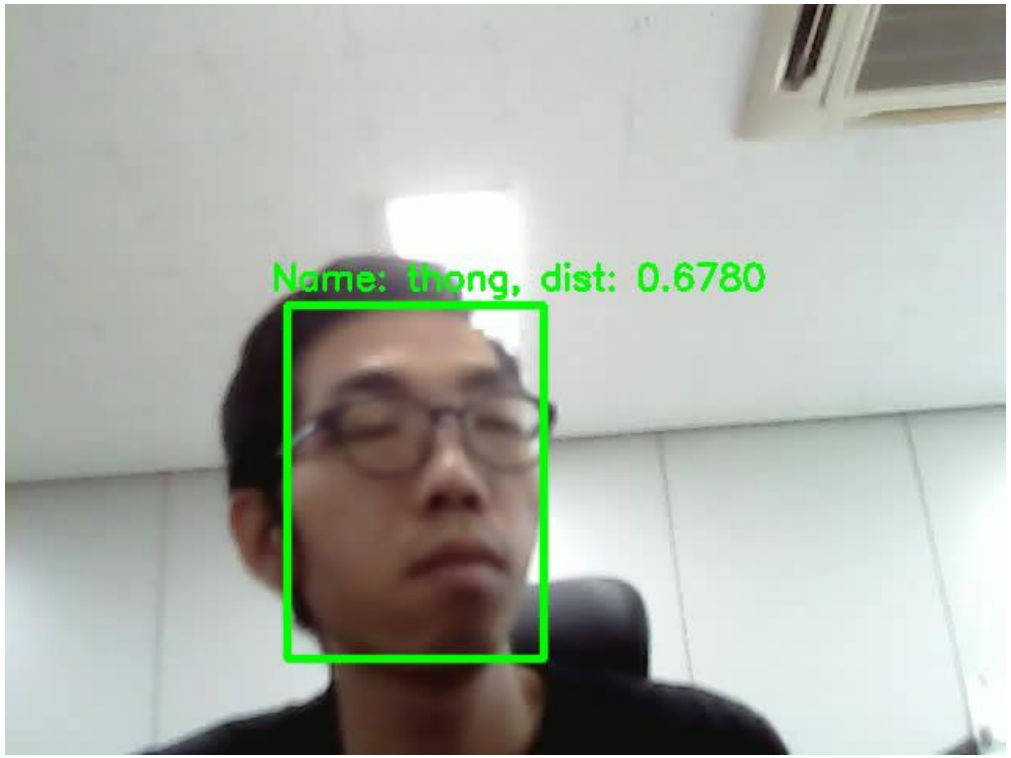


$$-1 \le S_C \le 1$$

- $S_C = -1$: $\theta = 180^o$, A and B are totally different.
- $S_C = 1$: $\theta = 0^o$, A and B are identical.

⇒ Larger *cosine similarity* is better.

Face recognition on new users



Not included left face, right face, ... images

Included left face, right face, ... images

Face recognition on new users

Photo requirements of new user for updating database: at least 5 photos from 5 different view as below figure, more view ⇒ more accurate.

Look straight

Turn right (left view)

Turn left (right view)

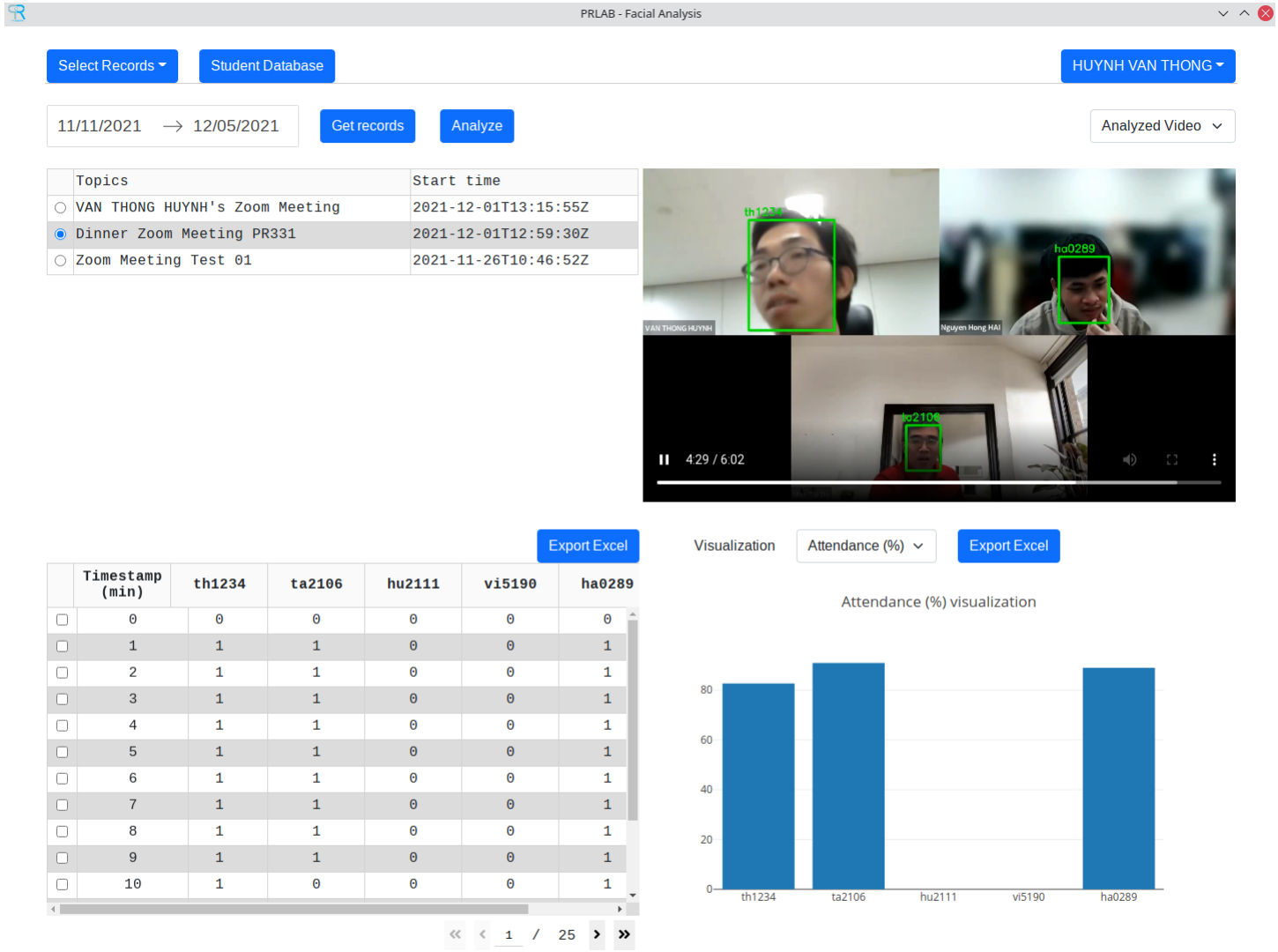
Look up

Look down

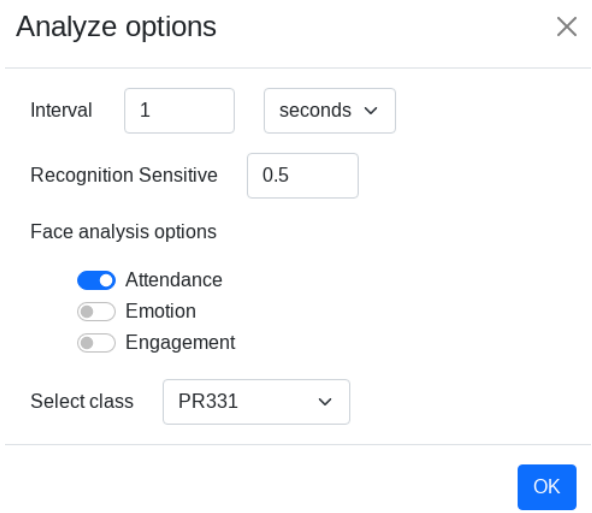
Optional

Optional

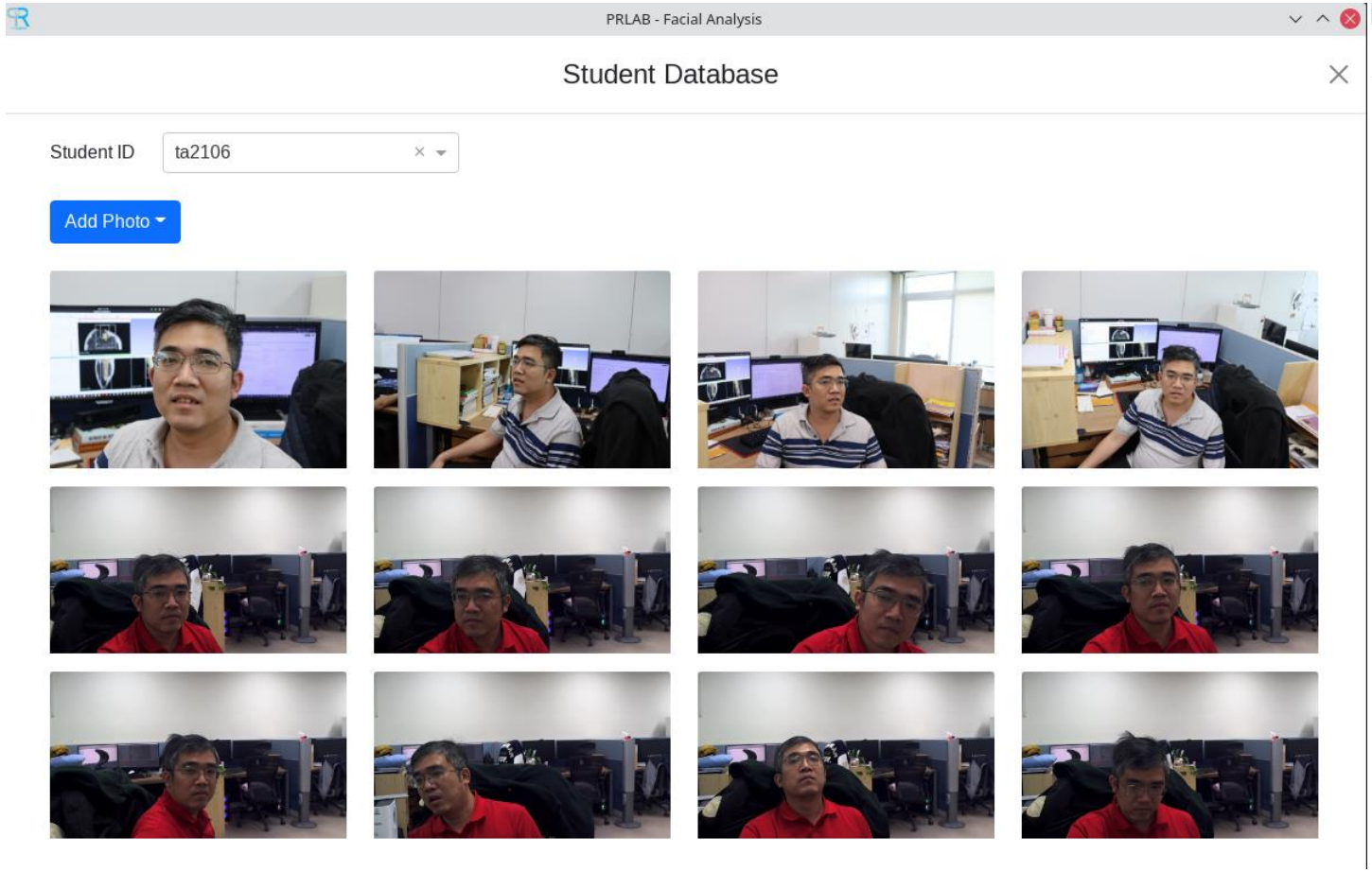
System implementation

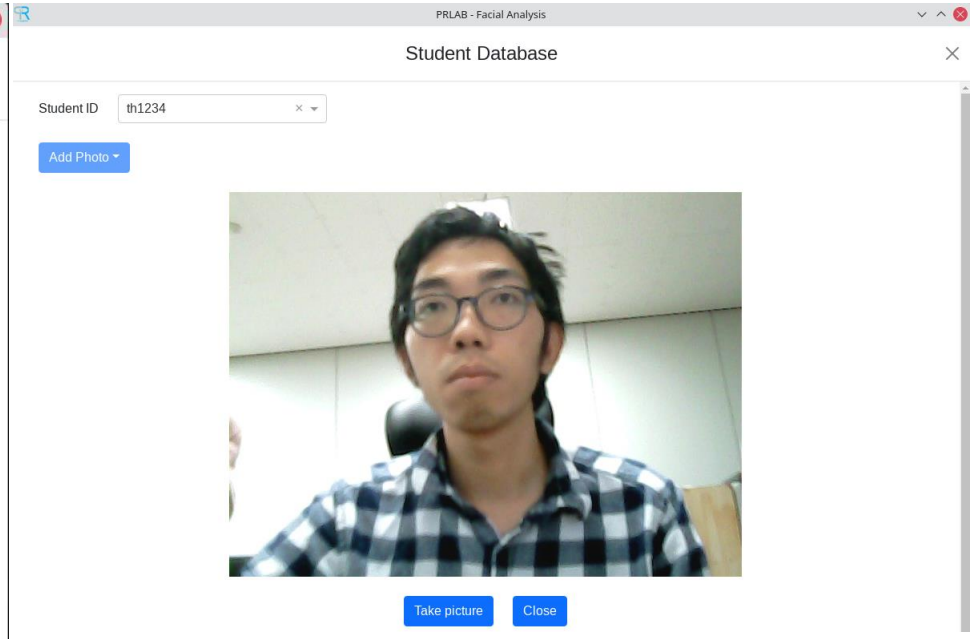


- ✓ **Select Records**: Zoom cloud record, local video, webcam.
- ✓ **Student Database**: add photo with webcam, or upload from PC.
- ✓ **Results**: export to excel, visualize by chart.



System implementation





Frameworks:

- ❑ User interface: Dash 2.0, Dash Bootstrap Components, PyQt6
- ❑ Face recognition models: Keras_insightface and deepinsight/insightface

Summary

- Can not recognize occluded faces (e.g., face with mask – need to capture that image).
- User interface: can not delete user, or photo of user in database. We need to do that manually by using Windows Explorer.
- Need more charts for visualization and data summary.