UATTENDED

UAttended

Attendance Checker Based On Face Recognition

Hsin-Ting Hsieh
Theresia Baier
Yuxin Wang

29/07/2020

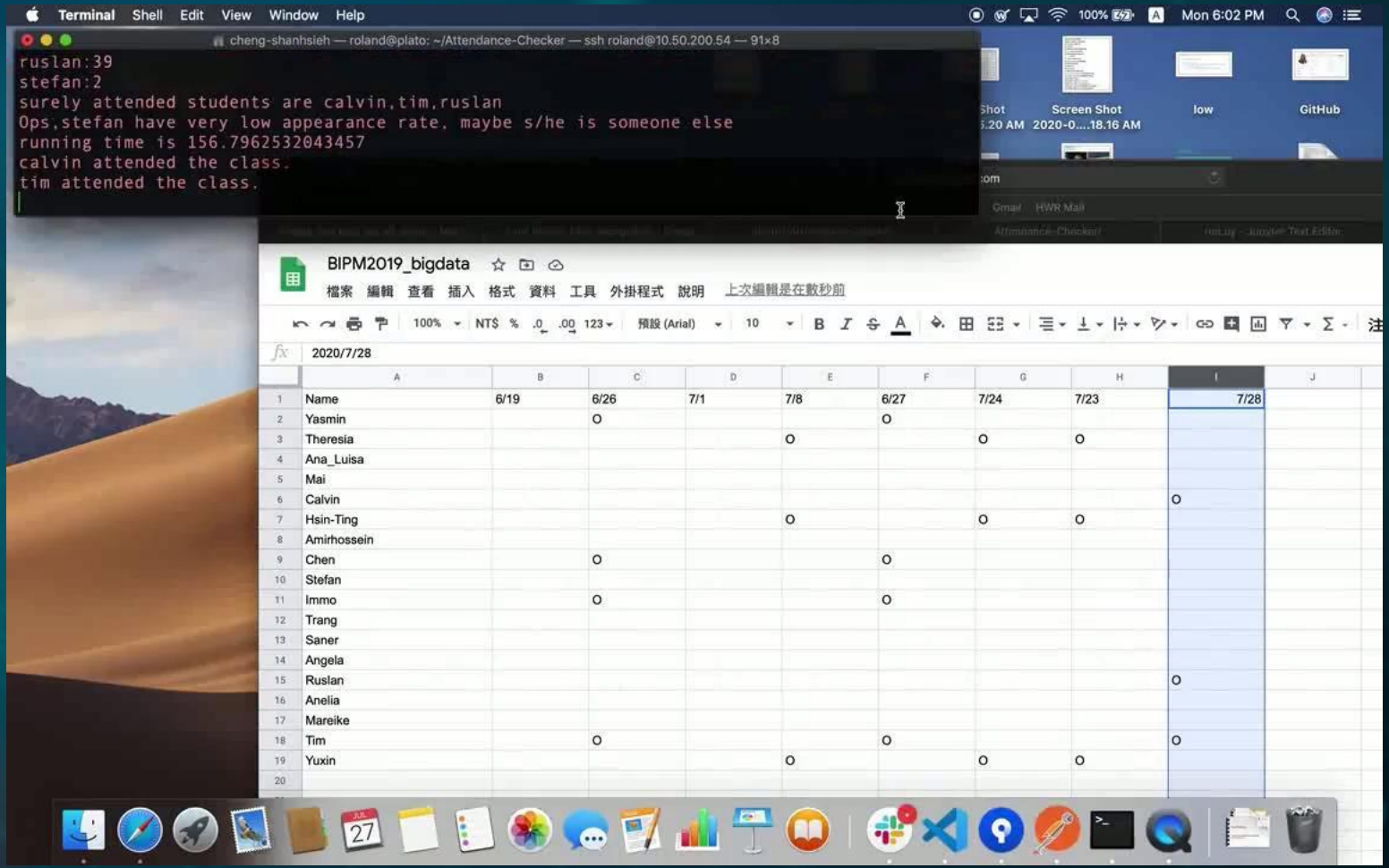
UATTENDED

Contents

Show Time! UAttended Demo

01

The user journey of generating an attendance list



	A	B	C	D	E	F	G	H	I	J
1	Name	6/19	6/26	7/1	7/8	6/27	7/24	7/23	7/28	
2	Yasmin		O			O				
3	Theresia				O		O	O		
4	Ana_Luisa									
5	Mai									
6	Calvin								O	
7	Hsin-Ting				O		O	O		
8	Amirhossein									
9	Chen		O			O				
10	Stefan									
11	Immo		O			O				
12	Trang									
13	Saner									
14	Angela									
15	Ruslan								O	
16	Anelia									
17	Mareike									
18	Tim		O			O			O	
19	Yuxin				O		O	O		
20										

Motivation

Attendance Check Is Everywhere

- An important action to evaluate students' dedication, the popularity of lectures/events
Schools, companies, events all need attendance check

Online Learning Rises Up Dramatically

- Over 1 billion students (pre-primary to tertiary) were affected due to COVID-19
- More and more distant learning programs

Filling-in Attendance List Is Forgettable

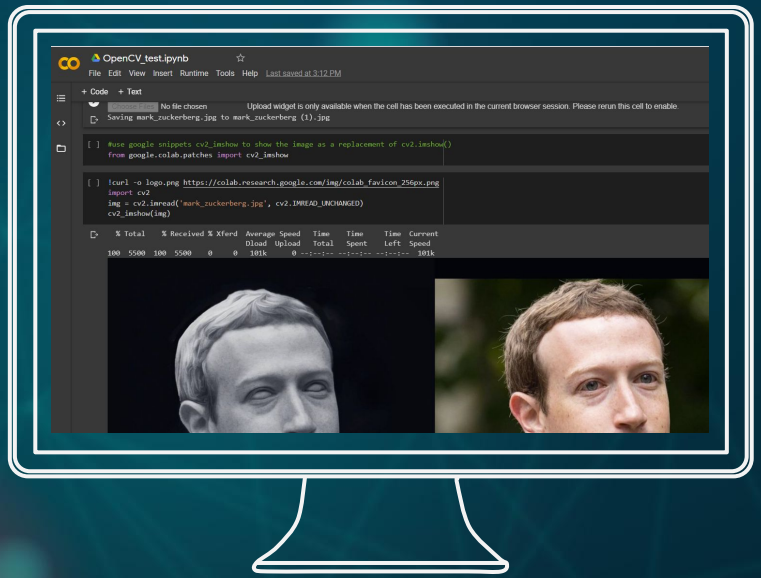
- Check-in in Excel every time for each lecture adds administrative efforts to professors and students
- Too many topics to focus on to remember check-in every time

Techniques & Methodology

02

Face recognition methods & tools, GPU, API

Working Environment



Google Colaboratory — **Development Environment**

Language — Python

GPU
16GB memory — **Processor**

Repository — GitHub

Face Recognition

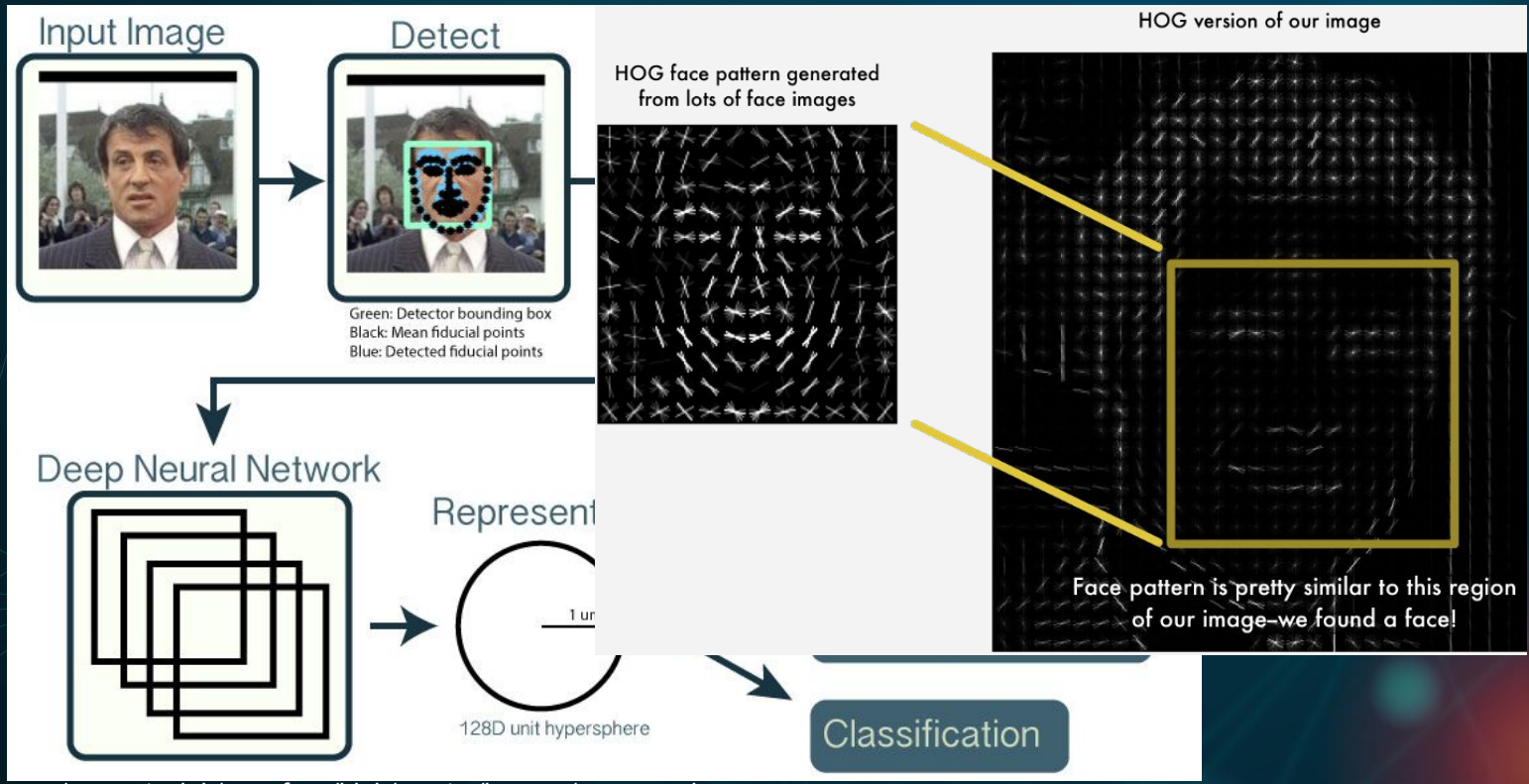


Photo sources: https://github.com/cmusatyalab/openface/blob/master/images/summary.jpg
https://miro.medium.com/max/875/1*6xgev0r-qn4oR88FrW6fiA.png

Computer Vision Library - OpenCV

Cascades

- About 6,000 classifiers for a face & thousands of data blocks
- Break the detection into 30 to 50 stages
- Only go to the further stage for detection if the block pasts the current stage

Image Capture & Detection

- Google snippets for cv2_imshow() is needed
- Load Cascades xml and apply
- Draw a rectangle around the face

Video Capture

- Via webcam or video files capture the video frame by frame.

OpenCV Face Detection Result

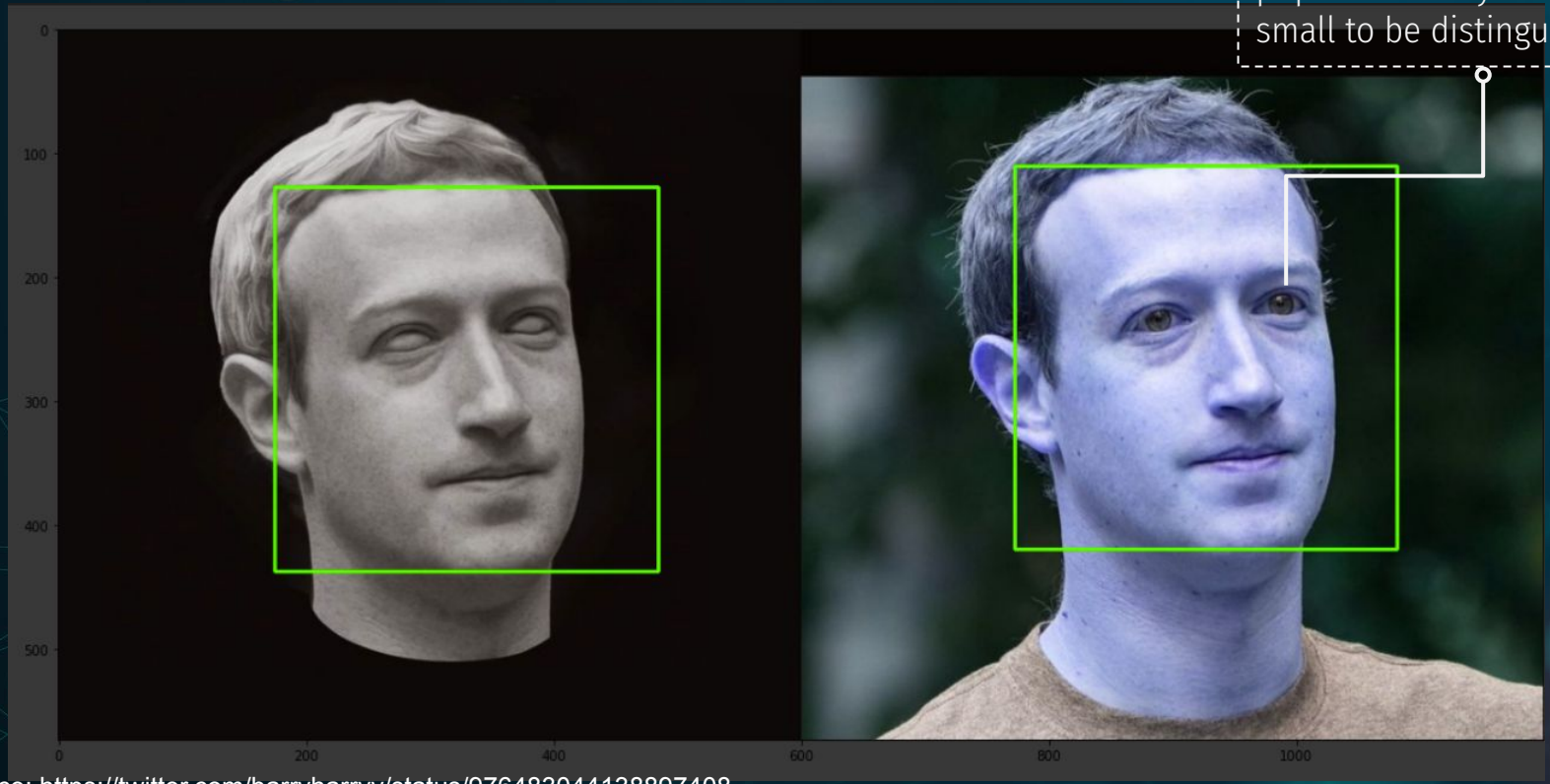


Photo source: https://twitter.com/barryharryy/status/976483044138897408

Computer Vision Library - OpenCV

Cascades

- About 6,000 classifiers for a face & thousands of data blocks
- Break the detection into 30 to 50 stages
- Only go to the further stage for detection if the block pasts the current stage

Image Capture & Detection

- Google snippets for cv2_imshow() is needed
- Load Cascades xml and apply
- Draw a rectangle around the face

Video Capture

- Via webcam or video files capture the video frame by frame.

Google Sheets API

- Google API library (291 public available)
- Authentication and authorization
- Wide range of functionalities

```
values = [
    [
        # Cell values ...
    ],
    # Additional rows ...
]
body = {
    'values': values
}
result = service.spreadsheets().values().update(
    spreadsheetId=spreadsheet_id, range=range_name,
    valueInputOption=value_input_option, body=body).execute()
print('{0} cells updated.'.format(result.get('updatedCells')))
```

Photo source: https://sheetsu.com/sheets-python

gspread

- **A Python API for Google Sheets.**
  - Integrate with pandas and Numpy packages

Quick Example

```
import gspread

gc = gspread.service_account()

# Open a sheet from a spreadsheet in one go
wks = gc.open("Where is the money Lebowski?").sheet1

# Update a range of cells using the top left corner address
wks.update('A1', [[1, 2], [3, 4]])

# Or update a single cell
wks.update('B42', "it's down there somewhere, let me take another look.")

# Format the header
wks.format('A1:B1', {'textFormat': {'bold': True}})
```

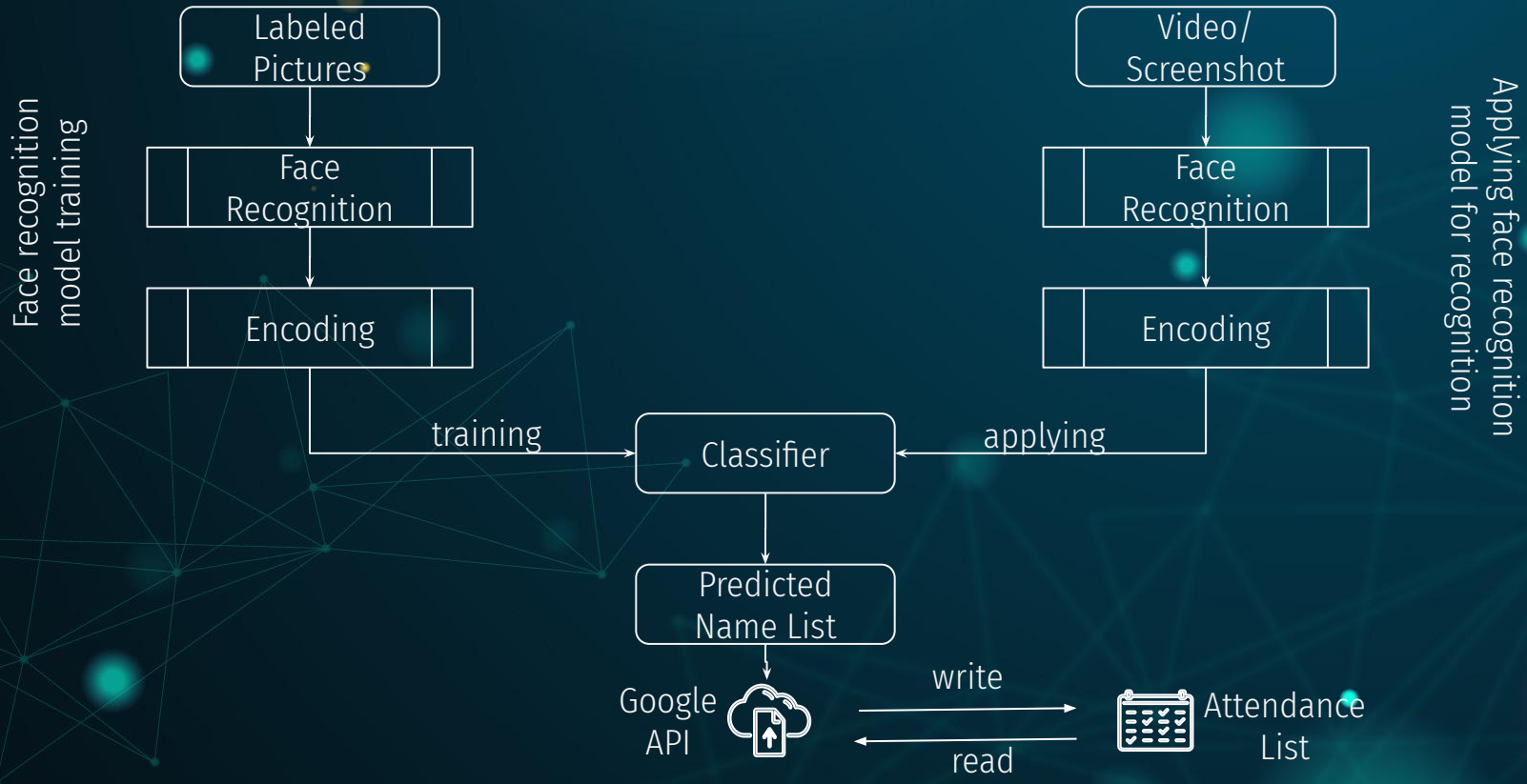
Photo source: https://sheetsu.com/sheets-python

Implement-ation

03

Preparation, training, application, evaluation, deployment

UAttended Architecture



Preparation

```
Structure:
        <test_image>.jpg
        <train_dir>/
            <person_1>/
                <person_1_face-1>.jpg
                <person_1_face-2>.jpg
                .
                .
                <person_1_face-n>.jpg
           <person_2>/
                <person_2_face-1>.jpg
                <person_2_face-2>.jpg
                .
                .
                <person_2_face-n>.jpg
            .
            .
            <person_n>/
                <person_n_face-1>.jpg
                <person_n_face-2>.jpg
                .
                .
                <person_n_face-n>.jpg
```

Photo source: https://github.com/ageitgey/face_recognition/blob/master/examples/face_recognition_svm.py

Training Face Recognition Model

Face Locations

Finding face in image for each person

Face Encodings

For images portraying only one person

SVC Classifier

Trained with face encodings and names

Testing Evaluation

01
Testing with Individual Pictures

	precision	recall	f1-score	support
amirhossein	1.00	1.00	1.00	9
ana_luisa	1.00	1.00	1.00	11
anelia	1.00	1.00	1.00	9
angela	1.00	1.00	1.00	5
calvin	1.00	1.00	1.00	8
chen	1.00	1.00	1.00	7
hsin-ting	1.00	1.00	1.00	5
immo	1.00	1.00	1.00	1
mai	1.00	1.00	1.00	3
mareike	1.00	1.00	1.00	10
ruslan	1.00	1.00	1.00	6
saner	1.00	1.00	1.00	1
stefan	1.00	1.00	1.00	3
theresia	1.00	1.00	1.00	5
tim	1.00	1.00	1.00	4
trang	1.00	1.00	1.00	4
yasmin	1.00	1.00	1.00	4
yuxin	1.00	1.00	1.00	8
accuracy	100 %		1.00	103
macro avg	1.00	1.00	1.00	103
weighted avg	1.00	1.00	1.00	103

02
Random Tests with Class Pictures

class (11) true	detected once	class (11) prediction
tim	TRUE	hsin-ting
theresia	TRUE	calvin
ana_luisa	FALSE	ana_luisa
amirhossein	TRUE	theresia
ruslan	FALSE	yuxin
angela	FALSE	mareike
immo	TRUE	chen
chen	TRUE	stefan
mareike	TRUE	ana_luisa
saner	FALSE	immo
hsin-ting	TRUE	tim
stefan	TRUE	mai
yuxin	TRUE	amirhossein
calvin	TRUE	
yasmin	FALSE	
mai	TRUE	
16	total	13
4	wrong/missing	
75,00%	**accuracy**	

Mean accuracy across four examples

88.84 %

Scenarios

01 Recognize with Screenshot

Pros: Fast and simple
Cons: need to choose the proper moment in case some students are late

02 Recognize with Videos/Webcam

Pros: No interruption during the lecture, can extract more info, not miss late students
Cons: Complicated computation, privacy concern

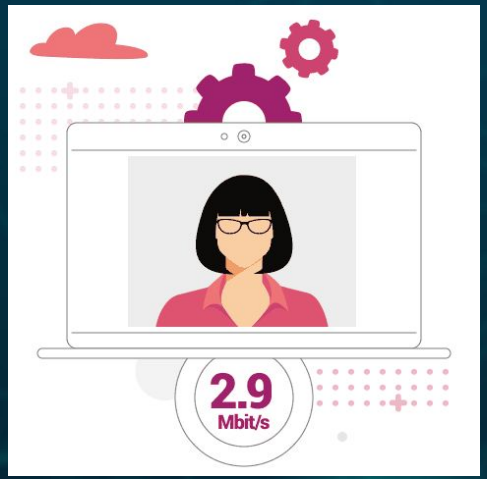


Photo source: https://www.eyeson.com/

Face Recognition Application

Image Capture

Catch faces based on the uploaded screenshot or video

Detect the locations of faces and generate encodings respectively

Face Location & Encoding

Face Recognition

Apply the classifier model to recognize input faces and generate a name list

Identify potential wrongly recognized faces based on defined threshold

Result Evaluation

Google API Connection

Authenticate Google API to connect the result with the attendance list sheet

Result Evaluation

01
Recognize with Screenshot

identify names that appear more than once

Calculate the percentage of names that could be wrongly recognized are calculated

```
Number of faces detected:  13
students attended are
anelia,saner,calvin,theresia,yuxin,hsin-ti
ng,ruslan,
angela,chen,immo,amirhossein,tim,stefan.
Ops! Someone looks too similar to
ana_luisa and was wrongly recognized!
7.6% of faces are recognized duplicately.
running time is 0.008185625076293945.
```

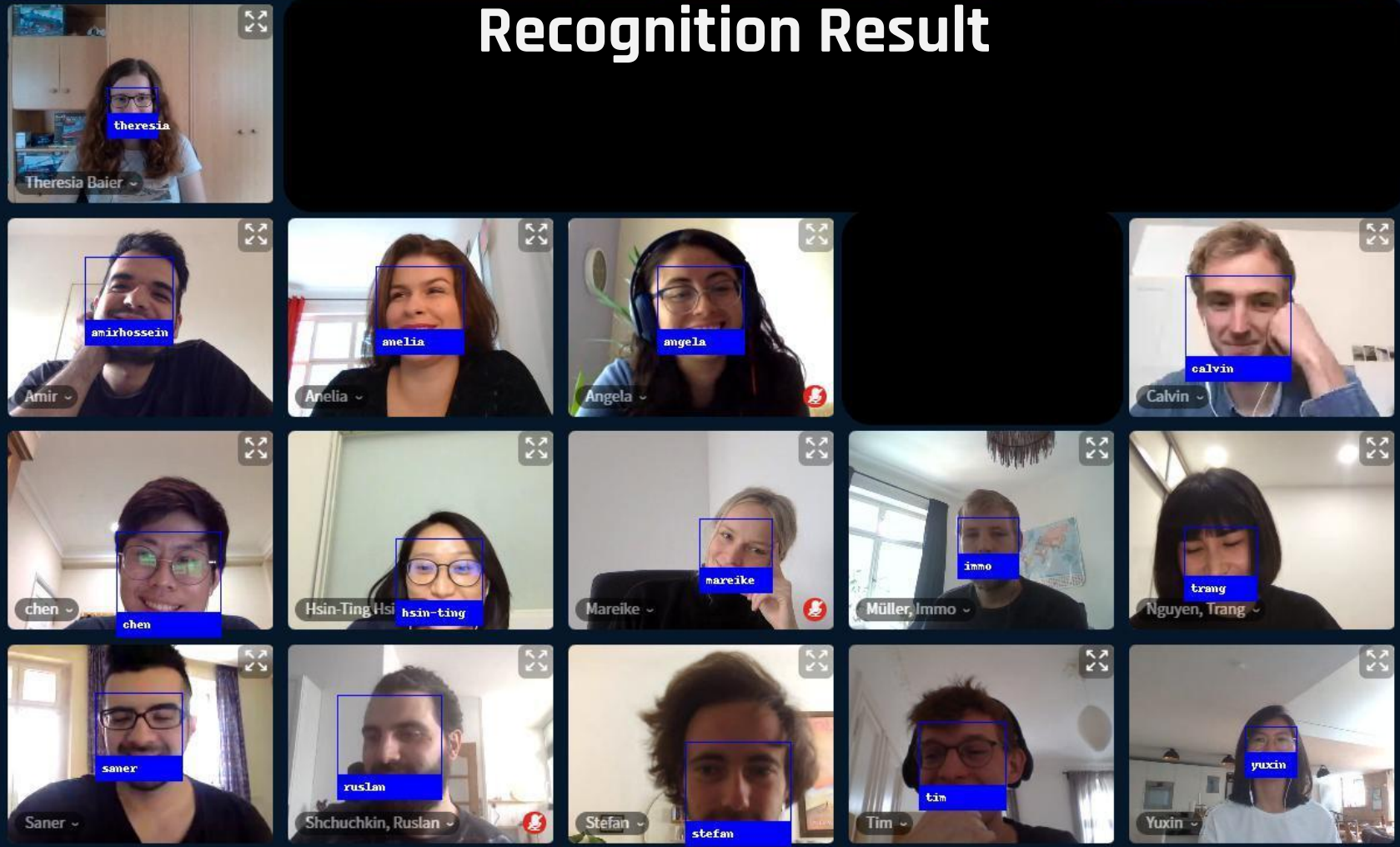
02
Recognize with Videos/Webcam

Generate a dictionary {key =name: value = the frequency appearing in all frames}

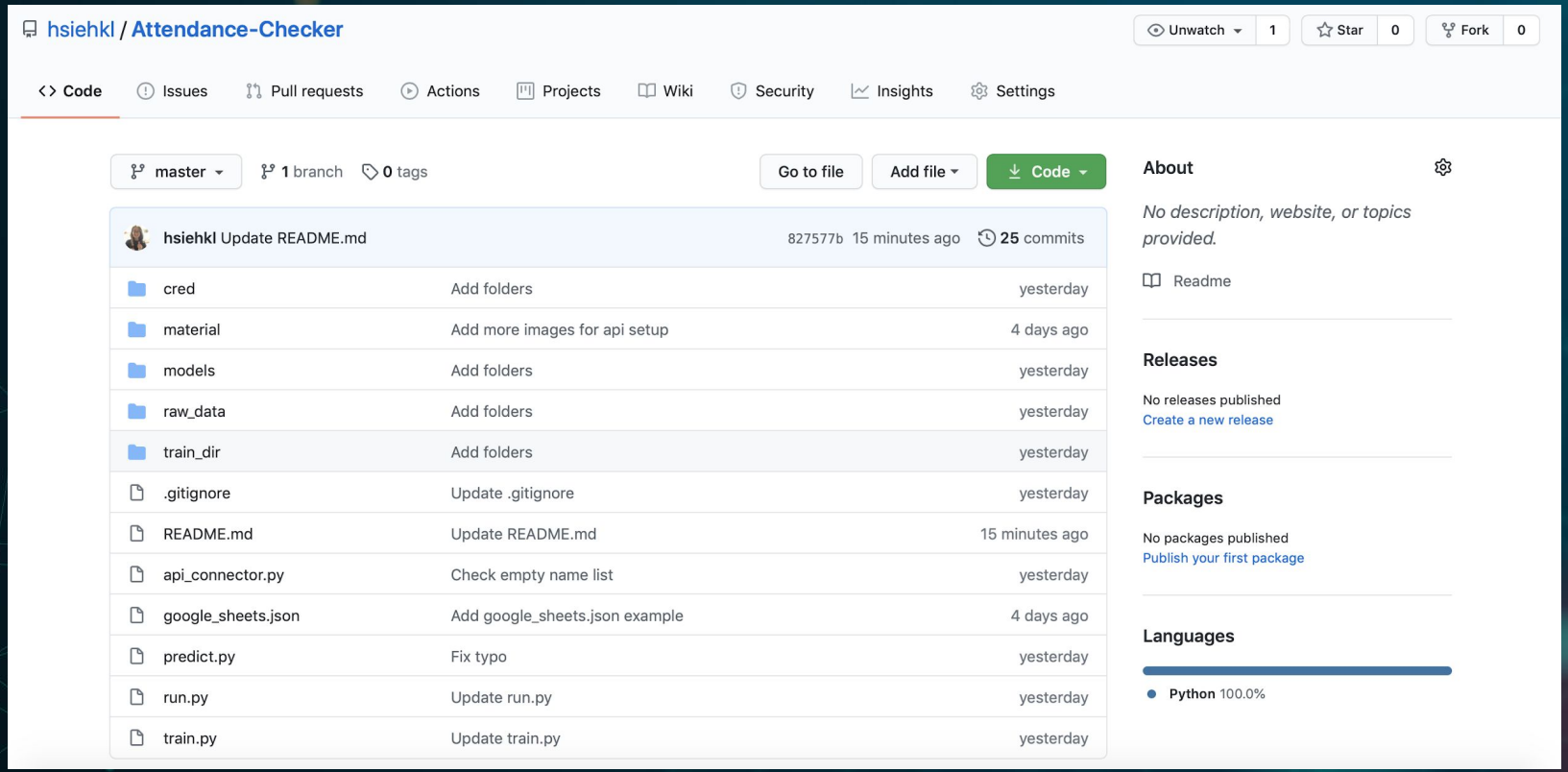
Threshold:
if the frequency of a name(vi)/max(v) <10% the name could potentially be wrongly recognized.

```
The frequency of names in all frames are:
yuxin: 63
hsin-ting: 89
theresia 89
ana_luisa: 2
Surely attended students are
yuxin,hsin-ting, theresia.
Ops, ana_luisa has very low appearance
rate, maybe s/he is someone else!
```

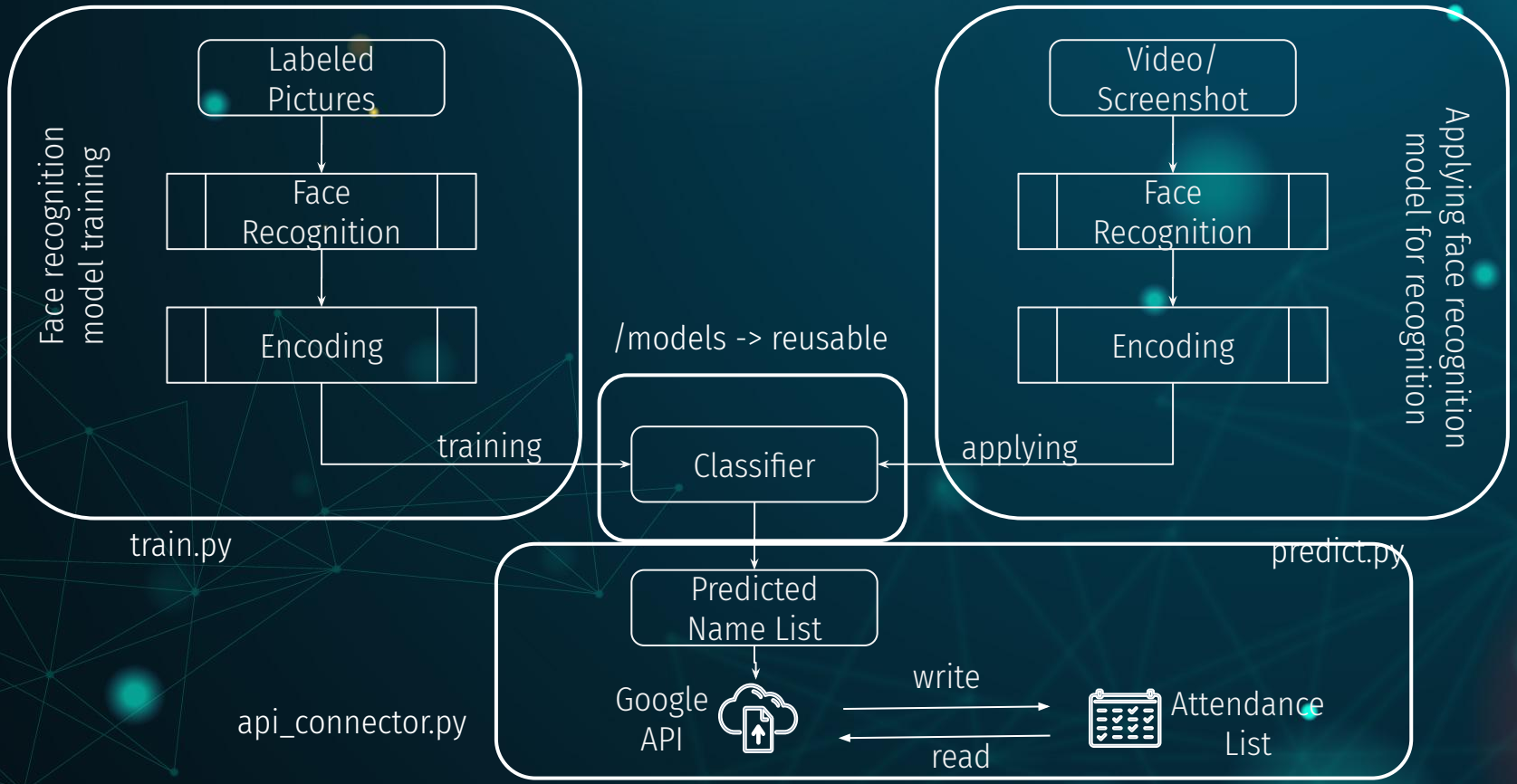
Recognition Result



UAttended Deployment on GitHub



UAttended Main Components



UAttended Deployment on GitHub

https://github.com/hsiehkl/Attendance-Checker

Conclusion & Outlook

04

Summary, reflection & further work

LET AI DO THE JOB

Challenges

- **Adaption To Google Colaboratory**
  File reading
  Display the video and image
- **Limited GPU memory**
  Can’t reliably apply CNN model
- **<100% Accuracy**
  Fairness
  Subtle manual adjustments needed
  Unknown faces can’t be recognized as unknown
- **Data privacy**
  Dissent of students

Misclassification

```
[8]  # Load the test image with unknown faces into a numpy array
     test_image = face_recognition.load_image_file(data_path_in + '/class/obama.jpg')

     # Find all the faces in the test image using the default HOG-based model
     face_locations = face_recognition.face_locations(test_image)
     no = len(face_locations)
     print("Number of faces detected: ", no)

     # Predict all the faces in the test image using the trained classifier
     name_list = []
     print("Found:")
     for i in range(no):
         test_image_enc = face_recognition.face_encodings(test_image)[i]
         name = model.predict([test_image_enc])
         name_list.append(*name)
         print(*name)
     print(name_list)
```

```
Number of faces detected:  1
Found:
yasmin
['yasmin']
```

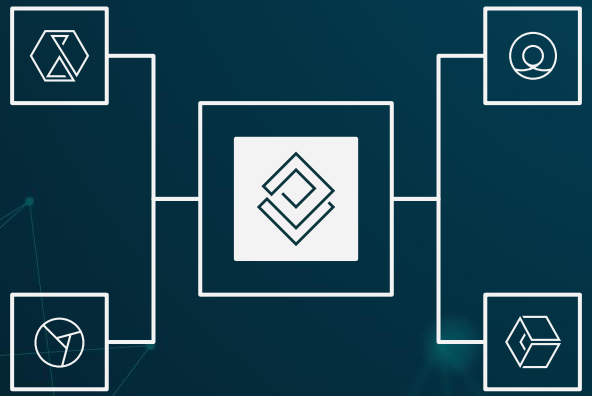
Photo source: https://raw.githubusercontent.com/ageitgey/face_recognition/master/examples/obama.jpg

Further Work

Get more powerful GPU
Apply CNN face recognition model
Tune classifier model
Reach 100% accuracy rate

Model Optimization

More Functions
Capture check-in and check-out time
Emotion Analysis
Lecture quality analysis



More user-friendly
Use the service directly without the necessity to download specific packages

Frontend

Privacy Security
Strategy and usage rules to ensure the secure environment and privacy protection

THANKS!

UATTENDED

Do you have any questions ?

Sources

Summary of Face Recognition Steps: Amos, B. (2015, September 24). Summary [Digital image]. Retrieved July 27, 2020, from https://github.com/cmusatyalab/openface/blob/master/images/summary.jpg

HOG Face Patterns: [HOG Face Patterns]. (n.d.). Retrieved July 27, 2020, from https://miro.medium.com/max/875/1*6xgev0r-qn4oR88FrW6fiA.png

Folder Structure: Baburajan, A. (2019, December 03). Face_recognition/face_recognition_svm.py at master · ageitgey/face_recognition. Retrieved July 26, 2020, from https://github.com/ageitgey/face_recognition/blob/master/examples/face_recognition_svm.py

Obama: Obama [Digital image]. (n.d.). Retrieved July 26, 2020, from https://raw.githubusercontent.com/ageitgey/face_recognition/master/examples/obama.jpg