淡江大學

資訊創新與科技學系

110學年度專題實作書面報告

視線校正

Eye Redirection

學生：	鄭明佐	408850385
	徐雅玟	408850351
	吳承恩	408860566

指導教授：　朱留　博士

中華民國 111 年 1 月

摘要

在視訊會議中，或者是在螢幕面前視訊與他人聊天時，是否有過感覺對方沒有在看自己的經驗？這是因為使用者必須得盯著攝像頭看，畫面看起來才會是有「眼神交流」的，但一般人在交談時並不會對著攝像頭講話，而是看著螢幕中的對方，然而當使用者看著螢幕裡的談話對象時，攝像頭拍攝到的畫面，卻又是沒有看著對方的畫面。因此我們希望可以透過程式，在使用者不用看著攝像頭的情況下，也能夠將視線修正過來。我們會以筆記型電腦上的攝像頭進行實作，在編程上搭配 OpenCV 來協助我們完成目標。

Abstract

In a video conference, or when chatting with others in front of the screen, have you ever felt that the other party is not watching you? This is because the user has to stare at the camera so that the screen looks like "eye contact", but most people do not talk to the camera when talking, but look at the other person on the screen. However, when using When the person is looking at the conversation partner on the screen, the image captured by the camera is not looking at the other person's image. Therefore, we hope that through the program, the user can correct the line of sight without looking at the camera. We will implement it with the camera on the laptop, and use OpenCV in programming to help us accomplish our goals.

Table of Contents

Chapter 1 Introduction

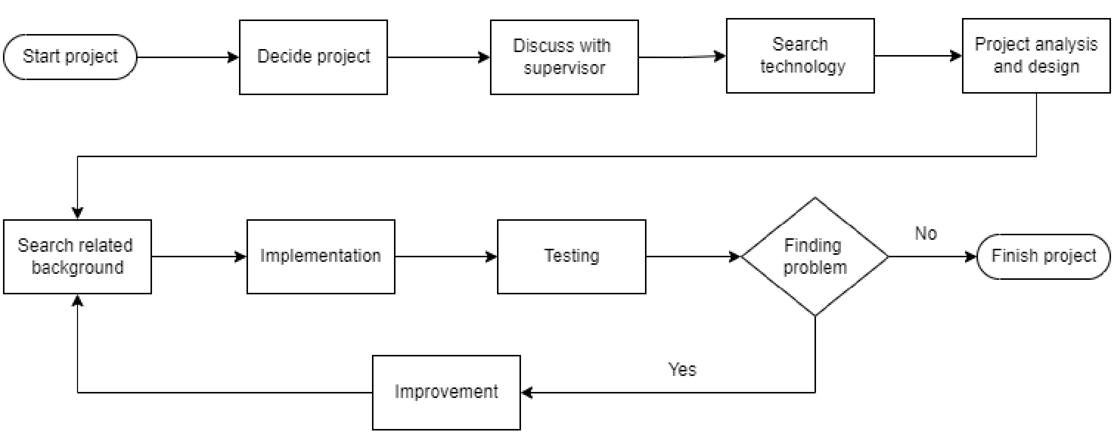
1.1 Motivation

In recent years, because the new crown pneumonia has spread globally, many governments have adopted strict measures such as lockdowns and social distancing in response to the impact of the epidemic. This has prevented many people from staying at home, but also caused many industries to be affected. The forced change to remote work has also led to numerous communication software such as Microsoft Teams, Google Meet, Zoom, and Discord, etc. have all emerged one after another. But when the use of communication software for remote meetings and remote teaching becomes our daily routine, time and space are compressed and efficiency is improved, but at the same time, it seems to have lost the interaction that should have been for face-to-face meetings and discussions. The most important issue is "Eye contact", when we realize that we are interacting with the person looking at us, this kind of eye contact will immediately cause a series of brain activities. However, our current long-distance video completely abandons this part of the communication, because when we video in front of the screen and look at the other person on the screen, the camera cannot capture that we are looking at the other person, and each other may feel that the other person is not there. Looking at ourselves makes us look weird in long-distance meetings and videos as if we are talking to the air. And our goal is to fix this error through a program so that we can look at the other person on the screen even if we are not looking at the camera but at the conversation partner on the screen.

1.2 Purpose

Our purpose is to improve the dilemma caused by incorrect eye gaze during the video, especially in today's day when video conferences are becoming more frequent, and similar technology is needed to correct the lack of eye contact during conversations between people.

1.3 Job Schedule



Chapter 2 Related Background

2.1 OpenCV

2.1.1 Introduction of OpenCV

OpenCV (Open Source Computer Vision Library) is a cross-platform computer vision library released open source based on the BSD (Berkeley Software Distribution) license terms. The project was initiated by Intel and can be used on multiple platforms such as Windows, Linux, Mac OS, ios, and Android. The core is written in C language, and APIs for multi-programming languages such as C++, Python, and Java are also provided. It has a rich library of commonly used image processing functions, which are mainly used for pattern recognition, image processing, and machine vision, and can quickly realize some image processing and recognition tasks.

2.1.2 Haar

Haar-like features are digital image features used to identify the existence of a specific object in an image. When detecting faces, we need to have such a sub-window that continuously shifts and slides in the image window to be detected. They look a bit like windows or lines or masks of various shapes. Every time the sub-window reaches a position, the feature value in the area will be calculated. Once the value has passed the set filter value, it can be judged as a face.

2.2 GAN

2.2.1 Introduction of GAN

GAN(Generative Adversarial Network) is an unsupervised machine learning method that learns by making two neural networks confront each other. These two neural networks are "Generator Network" and "Discriminator Network". Simply put, the generator is used to generate fake data that can fool the discriminator, and the discriminator tries to capture the fake data generated by the generator. The result of the confrontation between the two is the operating principle of its deep learning. At present, GAN is mostly used for data generation, such as image and audiovisual generation, synthesis, recognition, restoration, etc. However, GAN is divided into many types, and different GAN functions are also different. For example, WGAN, WGAN-GP, SN-GAN are used for training stability, and CycleGAN is used for image style conversion. And in our project, we use StyleGAN that can change photos or videos into another style.

2.2.2 StyleGAN

StyleGAN is one of GANs that can control the style and not create a style. StyleGAN designs the generator network structure and tries to control the image generation process: the generator starts from the learned constant input and adjusts the image "style" of each convolutional layer according to the latent code, thereby directly controlling the image characteristics; in addition, combining The noise directly injected into the network can change the random attributes, such as freckles, hair, in the generated image. StyleGAN can achieve unsupervised attribute separation to a certain extent, and perform some style mixing or interpolation operations.

2.3 Gaze Redirection

There is more than one way of gaze redirection, such as using GANs to synthesize images of eyes with redirected gaze or using segmentation masks.

Segmentation means, the difference between Semantic Segmentation and Instance Segmentation, and different Segmentation models like U-Net, Mask R-CNN, etc.

Chapter 3 Research Method

3.1 Research Information

After we decided that the topic is to adjust the view of the video, we first looked for multiview information on the Internet. After studying multiview, we still did not know how to implement our topic. We found a paper that explained how to detect the face and filter out the part of the eye through the method, and we kept searching for the steps after face detection.

We knew that we could use OpenCV and Dlib directly to implement face detection, and we also learned about the method of detecting faces through the author's related website, and then we decided to use the haar library in OpenCV to complete the part of face detection, and we learned that the topic we wanted to do was called gaze correction approach by our predecessors through searching data, and we found a paper after googling After searching for information, we found a paper with similar purpose, and decided to use StyleGAN to generate a new image, and then adjust the view by ternary mask.

3.2 Tool(The following tool different versions may have problems)

3.2.1 Hardware Requirement

According to the StyleGAN authors "one or more high-end NVIDIA GPUs with at least 11 GB of DRAM. we recommend the NVIDIA DGX-1 with 8 Tesla V100 GPUs".

3.2.2 OpenCV

We use Visual Studio to create the project, download OpenCV, and set the path and then we can execute the face detection part.

3.2.3 CUDA&CUDNN

We use CUDA 10.0 with the corresponding CUDNN version 7.4.

3.2.4 Python, Anaconda, TensorFlow

We use python 3.6 and the corresponding anaconda3 version 5.2.0. Then we create a virtual environment in Anaconda. Then we load TensorFlow-GPU version 1.13.1 and download StyleGAN author's code and run pretrained_example.py.

Chapter 4 Research Results and Discussion

4.1 Research Results and Discussion

The result of face detection is shown in Figure 1, the face can be detected smoothly under the condition of sufficient light and not wearing too much ornament.

We searched for information about StyleGAN and found that there were very few people doing this work. We tried to match the official version with the successful version on the Internet and the result was still similar to Figure 2.

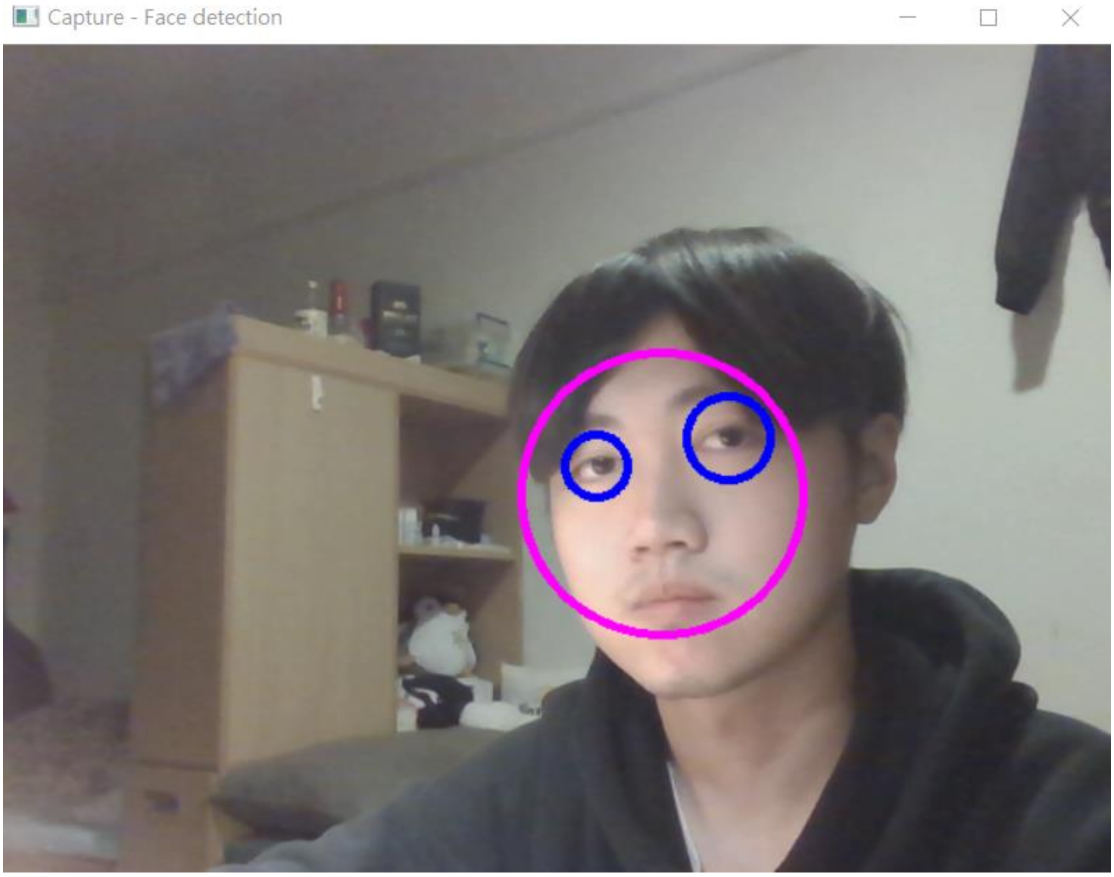
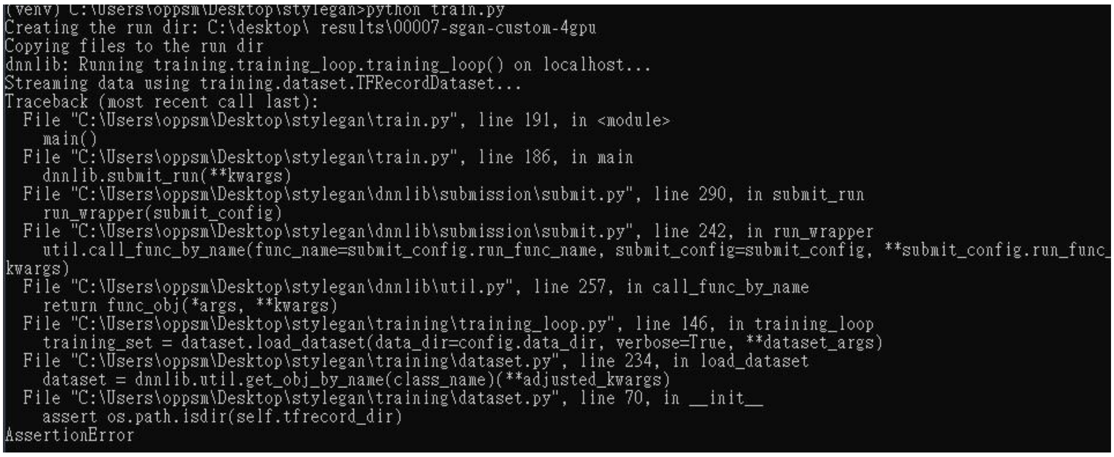


Figure 1 The result of face detect and eye capture



```
(venv) C:\Users\oppsm\Desktop\stylegan>python train.py
Creating the run dir: C:\desktop\ results\00007-sgan-custom-4gpu
Copying files to the run dir
dnnlib: Running training.training_loop.training_loop() on localhost...
Streaming data using training.dataset.TFRecordDataset...
Traceback (most recent call last):
  File "C:\Users\oppsm\Desktop\stylegan\train.py", line 191, in <module>
    main()
  File "C:\Users\oppsm\Desktop\stylegan\train.py", line 186, in main
    dnnlib.submit_run(**kwargs)
  File "C:\Users\oppsm\Desktop\stylegan\dnnlib\submission\submit.py", line 290, in submit_run
    run_wrapper(submit_config)
  File "C:\Users\oppsm\Desktop\stylegan\dnnlib\submission\submit.py", line 242, in run_wrapper
    util.call_func_by_name(func_name=submit_config.run_func_name, submit_config=submit_config, **submit_config.run_func_
kwargs)
  File "C:\Users\oppsm\Desktop\stylegan\dnnlib\util.py", line 257, in call_func_by_name
    return func_obj(*args, **kwargs)
  File "C:\Users\oppsm\Desktop\stylegan\training\training_loop.py", line 146, in training_loop
    training_set = dataset.load_dataset(data_dir=config.data_dir, verbose=True, **dataset_args)
  File "C:\Users\oppsm\Desktop\stylegan\training\dataset.py", line 234, in load_dataset
    dataset = dnnlib.util.get_obj_by_name(class_name)(**adjusted_kwargs)
  File "C:\Users\oppsm\Desktop\stylegan\training\dataset.py", line 70, in __init__
    assert os.path.isdir(self.tfrecord_dir)
AssertionError
```

Figure 2 Execution results of pretrain_example.py

Table 3 shows the version we have tried.

	Python	Anaconda3	TensorFlow	CUDA	CUDNN
1st	3.9	2021.11	2	11.5	8.3

2nd	3.7	2020.02	1.13	10.0	7.6
3th	3.6	5.2.0	1.13	10.0	7.4

Table 3 The three corresponding versions we have tried

Chapter 5 Review and Future Development

5.1 Review

Looking back on this project, we find out some problems. First of all, we were too late to start the project. Because we were too late to find the supervisor to discuss the topic of the project, the time and schedule that we can study and do the project are delayed. Although we find the supervisor to discuss the topic finally, we didn't evaluate our ability and the difficulty of the topic. We didn't think about the difficulty of the implementation before we decided to choose this project Based on the above reasons, we met a lot of challenges and had difficulties in implementing. However, we didn't ask professors for help. We just continuously searched the net to find the answer. Due to those reasons, we didn't finish the project.

After the end of this course Practice of Projects, although unfortunately, we were unable to complete the research within the time limit, after participating in this course, we continued to learn from the process, and we learned how to research projects, how to discuss and coordinate with team members and professors, and more importantly, we learned how to implement our problems and think about solutions in the process of achieving our goals. When researching Projects, we learned how to search for information on the Internet and obtain the information we want from the huge amount of data and integrate it.

We will learn from the mistakes made in this course and minimize the chances of them in the future. We will be more positive and cautious when we do other projects. Think ahead about the challenges we may meet and consider our ability and the difficulty of the project. Asking the professionals for help timely to make the project better.

5.2 Future Development

In future work, we plan to use a segmentation mask to finish gaze redirection, which will have to need knowledge about image recognition, also need to integrate the relevant programs, and then build into an app, it is recommended to have the relevant knowledge and then continue to study.

Chapter 6 Conclusion

6.1 Conclusion

We divide this project into two main parts, and the first part is face detection and eye capture. We first download OpenCV and set it up in Microsoft Visual Studio that we want to use, we use OpenCV Haar cascades to detect faces. The second part is the eye line correction. The second part is the eye view correction, we use StyleGAN in GAN to generate our new eye view, however, we encountered problems with the corresponding versions of Python, Anaconda, and TensorFlow in this part, so we were not able to finish the project as scheduled. In the future, we will continue to do the project. We will use segmentation mask to finish gaze redirection.

References

Harsimran Kaur. Roberto Manduchi(2021). Subject Guided Eye Image Synthesis with Application to Gaze Redirection. University of California, Santa Cruz.

https://openaccess.thecvf.com/content/WACV2021/papers/Kaur_Subject_Guided_Eye_Image_Synthesis_With_Application_to_Gaze_Redirection_WACV_2021_paper.pdf

https://www.tutorialspoint.com/how-to-track-the-eye-in-opencv-using-cplusplus

林國暐(2010)。智慧型人眼狀態偵測系統。龍華科技大學學報第三十期,13-26,桃園。