Image Processing

< Team Project 2 – Final Report >

Team #8 - I am Image

20190402 Lee Jaemin

20191127 Kim Doohong

20194921 Lee Giwoong

50231583 Ali Mhrez

1. Outline of program

Many people living in the modern society want to express themselves and communicate with others by sharing their daily lives, and at the center of it is social network services. Social network services called SNS are available in various types, such as YouTube and Instagram, and SNS users can show their photos or videos to others by uploading their own photos or videos. In this process, portrait rights infringement may occur when the face of another person other than your own is exposed, and to prevent this, users often mosaic and upload other people's faces. However, in the case of videos, it is quite difficult to mosaic themselves, and even in the case of photos, many users cannot mosaic themselves. Therefore, we tried to create a program that helps SNS users mosaic their faces they want to cover more conveniently and simply, and through this, we tried to protect the portrait rights of others.

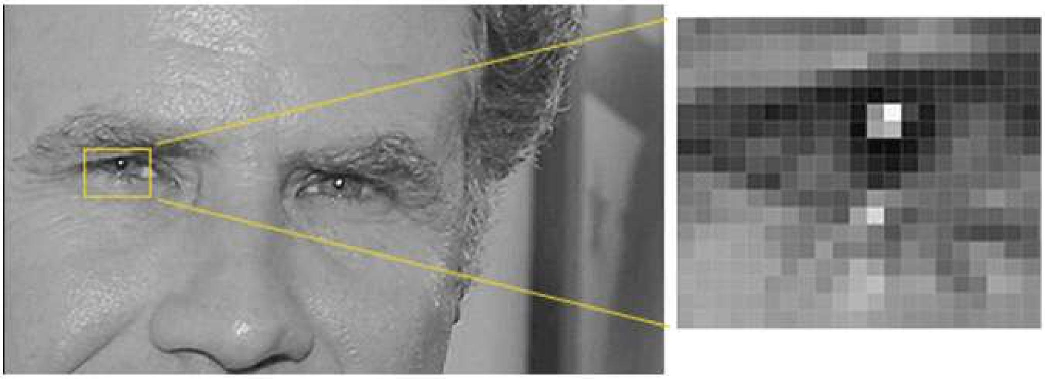
2. Design of program

There are two main features that this program supports: photo mosaic and video mosaic. Both functions work by selecting the target the user wants and mosaic everyone except the target. I will explain the technology used first and then explain the process of use.

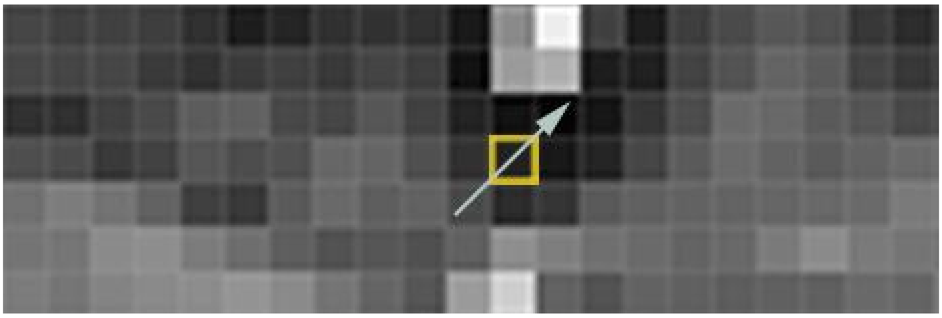
(Tech-1) Face recognition

- We tried to implement face recognition, but in order to implement face recognition directly, it was necessary to learn numerous faces using machine learning technology and required a high-quality understanding of machine learning. Also, in this project, which avoids opencv by using a lot of opencv, it is not suitable and we don't have enough time, so we used face_recognition.
- In order to find a face in an image, we turn the image into black and white because we don't need color data to find a face.

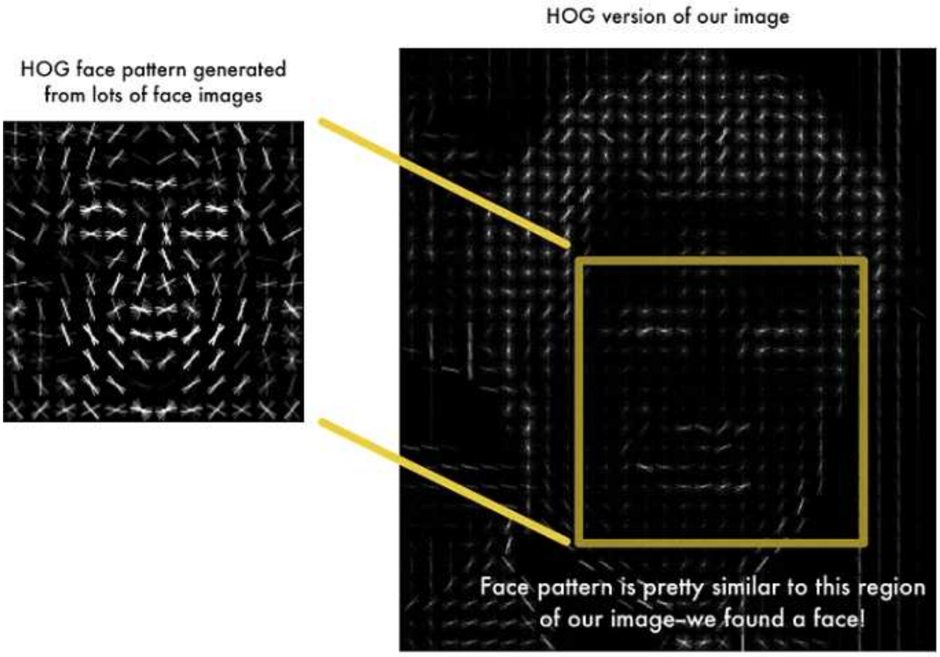
- Then, let's look at every single pixel in the image, one at a time. For every single pixel, we will also look at the pixels surrounding it directly.



- Our goal is to figure out how dark that pixel is compared to the pixels directly surrounding it. Through this, we want to draw an arrow to indicate the direction in which the image is darkened.



- Repeat this process for all pixels of the image, the eventually changes will change the arrow.These arrows are called graph (gadients) and can see the flow from bright parts from the entire image.
  As a result, we transform the original image into a very simple representation in which the basic structure of the face is presented in a simple way.
- What we need to do to find faces in this HOG image is to find the part in the image that looks most similar to the well-known HOG pattern extracted from many training face images.

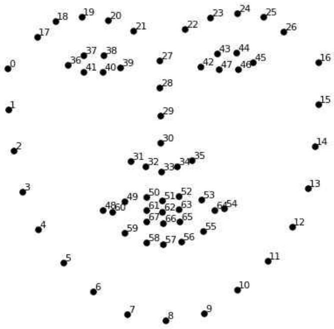

HOG version of our image
HOG face pattern generated from lots of face images
Face pattern is pretty similar to this region of our image–we found a face!

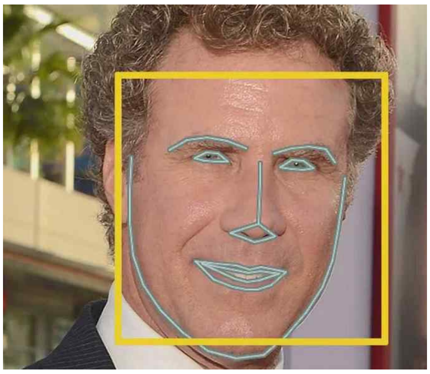
- Now, in order to recognize whether you are the same person, you must solve a new problem that makes your face look completely different to your computer if you look in a different direction.

  To solve this problem, twist each picture so that the eyes and lips are always in the standard position. This makes it easier to compare faces in the following steps.

  To do this, we use an algorithm called face landmark estimation.

- The basic idea is to find specific points called 68 landmarks that exist on every face. Now we are said to train machine learning algorithms so that we can find these 68 specific points on any face.

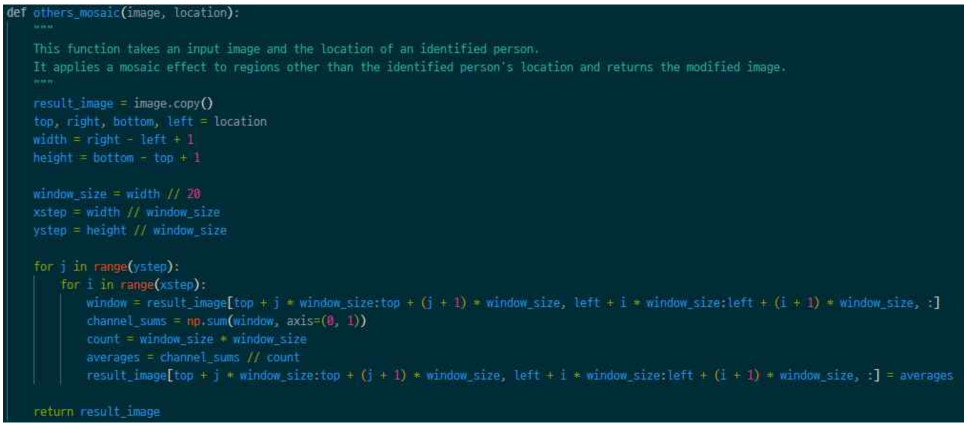




- This is the result of 68 facial landmarks displayed on our test images.

(Tech-2) Mosaic

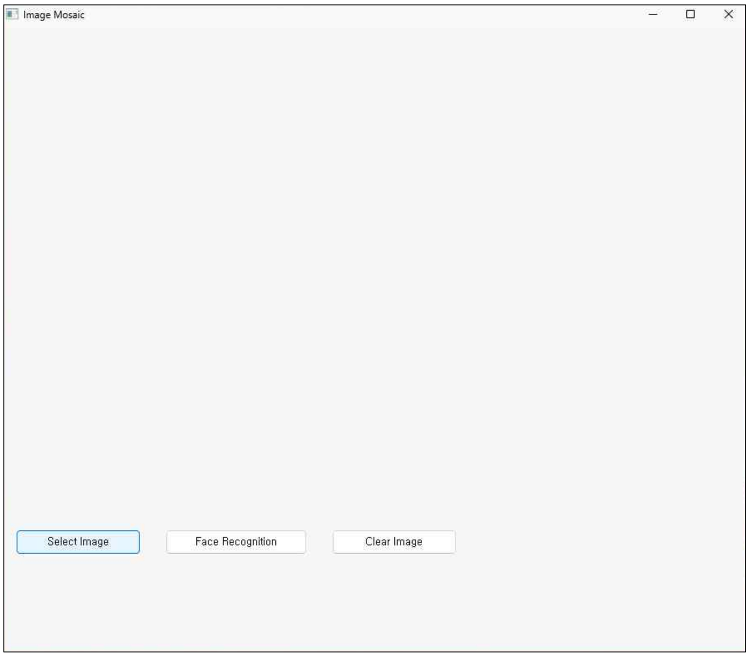
- First, the entire image and location value to be mosaic-processed are received.
- Then, a mosaic pixel size suitable for the face size is determined.
- Finally, the mosaic is processed by calculating the value using the average of the square values in a format similar to assignment 2.

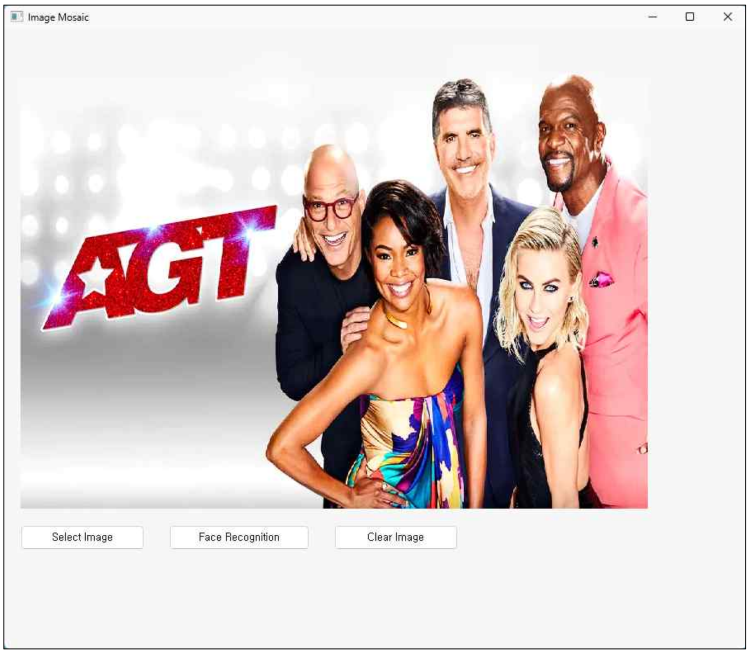


```
def others_mosaic(image, location):
    """
    This function takes an input image and the location of an identified person.
    It applies a mosaic effect to regions other than the identified person's location and returns the modified image.
    """
    result_image = image.copy()
    top, right, bottom, left = location
    width = right - left + 1
    height = bottom - top + 1

    window_size = width // 20
    xstep = width // window_size
    ystep = height // window_size

    for j in range(ystep):
        for i in range(xstep):
            window = result_image[top + j * window_size:top + (j + 1) * window_size, left + i * window_size:left + (i + 1) * window_size, :]
            channel_sums = np.sum(window, axis=(0, 1))
            count = window_size * window_size
            averages = channel_sums // count
            result_image[top + j * window_size:top + (j + 1) * window_size, left + i * window_size:left + (i + 1) * window_size, :] = averages

    return result_image
```

(Photo-1) Upload the image

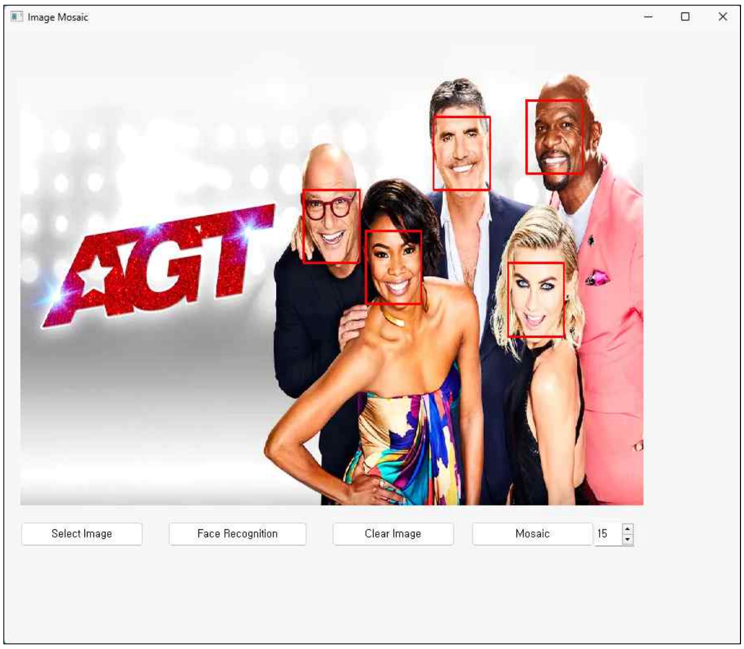
- First, upload a picture to the program using the “Select Image” button.





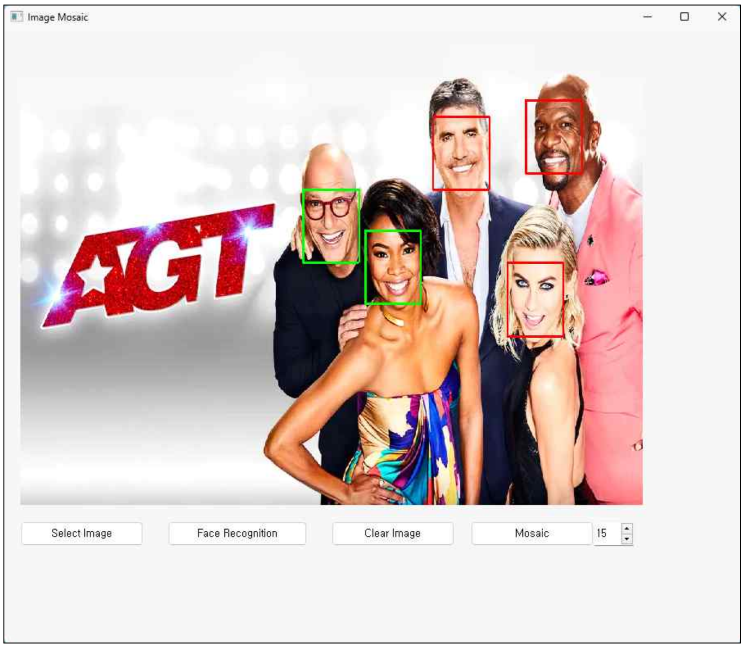
(Photo-2) Face Recognition

- After that, when the "Face Recognition" button is pressed, an area recognized as a human face in the picture is displayed in a red box.



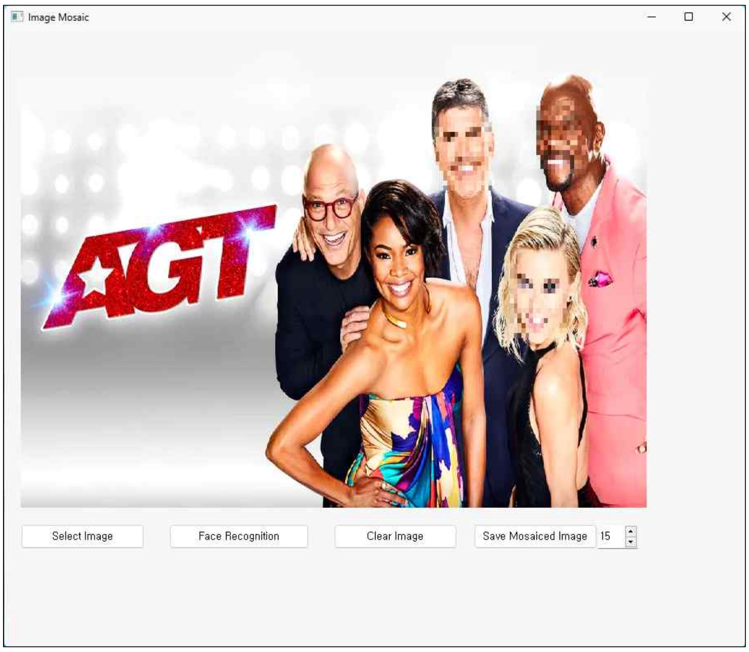
(Photo-3) Select the faces

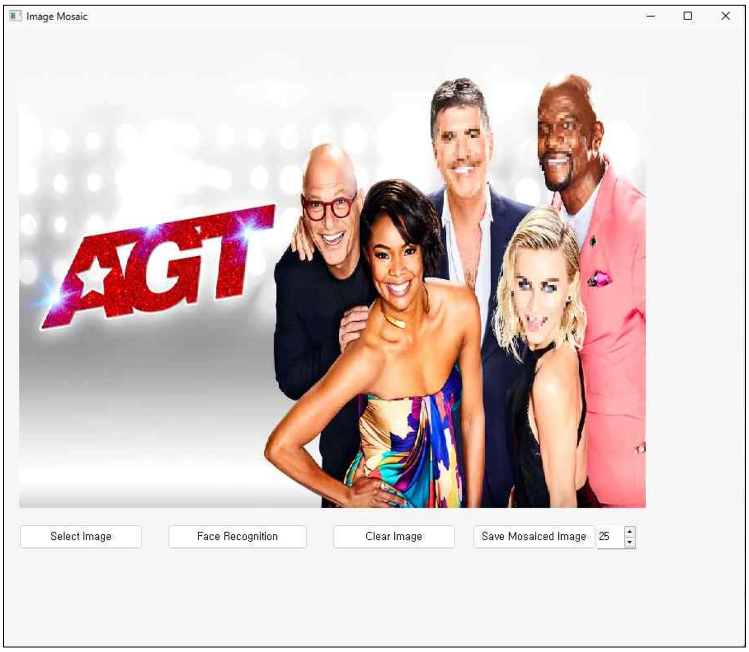
- Choose faces from the red boxes not to mosaic.



(Photo-4) Mosaic

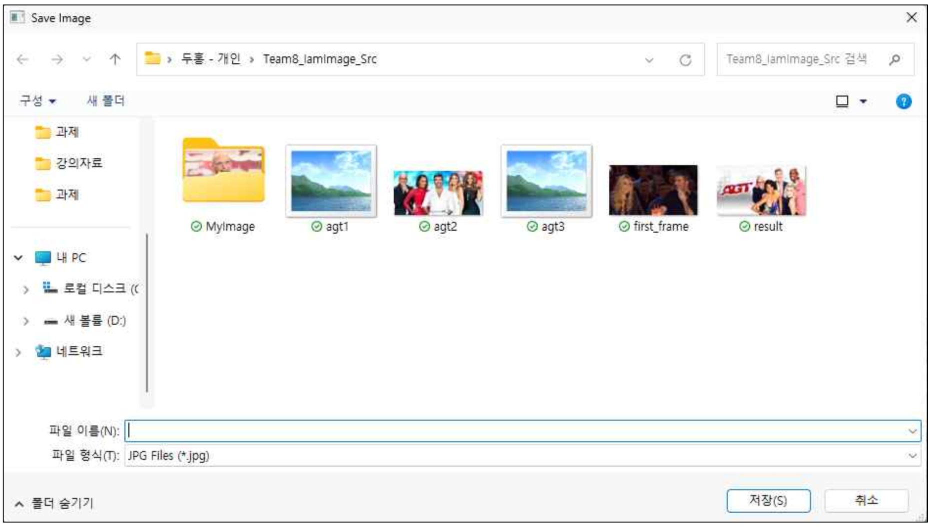
- You can mosaic unselected faces by pressing the "Mosaic" button. Here, you can set the mosaic intensity by modifying the number next to the button.
- Looking at the images below, it can be seen that the smaller the number, the stronger the mosaic, and the larger the number, the weaker the mosaic.





(Photo-5) Save the image

- If you press the "Save Mosaiced Image" button, the following window will appear and you can save it under your desired name.

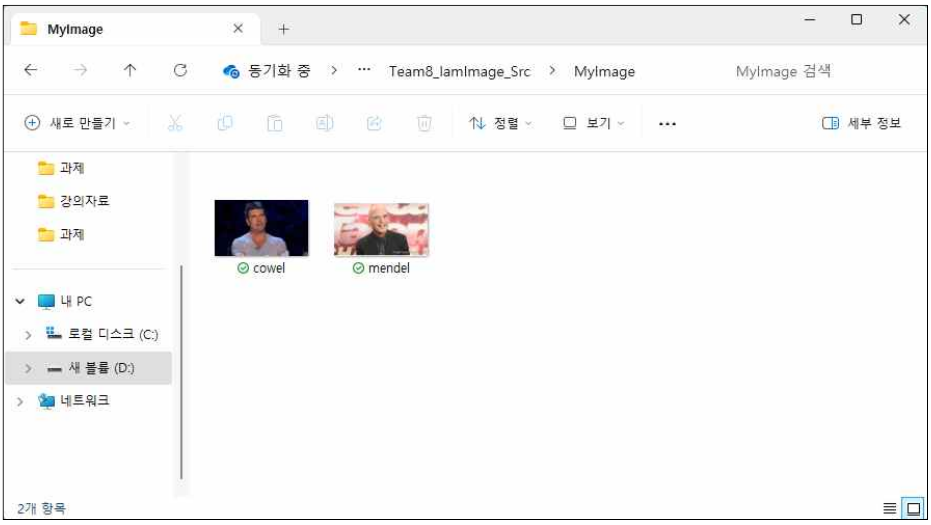


파일 이름(N):

파일 형식(T): JPG Files (*.jpg)

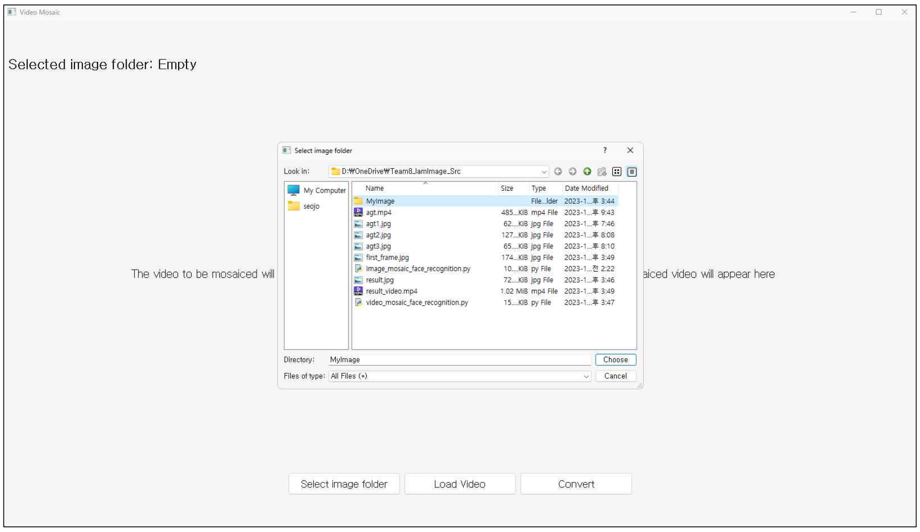
(Video-1) Prepare the image folder

- Prepare a folder with images of people who don't want me to mosaic.



(Video-2) Upload the video & image folder path

- Upload the video you're trying to mosaic and the path to the image folder you just prepared.
- Use the "Select image folder" button and the "Load Video" button.



Selected image folder: Empty

The video to be mosaiced will

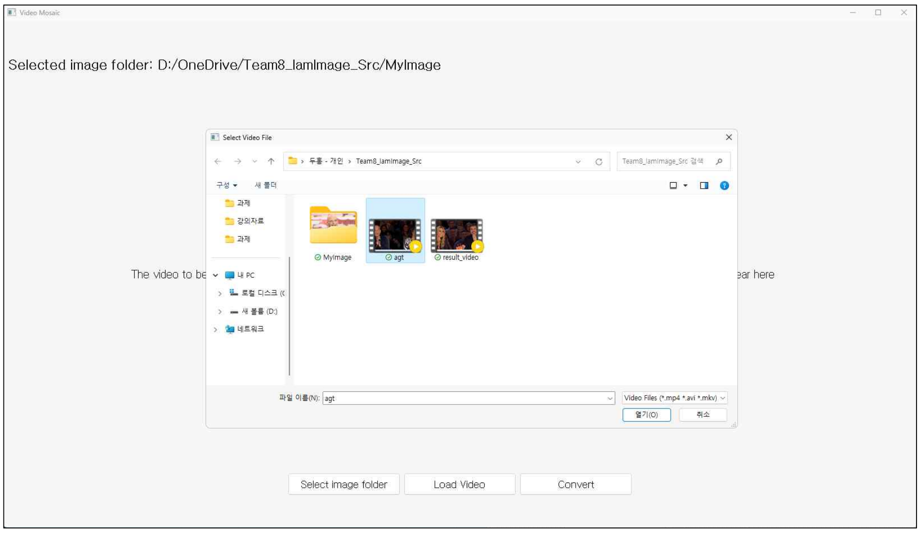
aiced video will appear here

Look in: D:₩OneDrive₩Team8_IamImage_Src

Name	Size	Type	Date Modified
MyImage		File...lder	2023-1...후 3:44
agt.mp4	485...KiB	mp4 File	2023-1...후 9:43
agt1.jpg	62....KiB	jpg File	2023-1...후 7:46
agt2.jpg	127...KiB	jpg File	2023-1...후 8:08
agt3.jpg	65....KiB	jpg File	2023-1...후 8:10
first_frame.jpg	174...KiB	jpg File	2023-1...후 3:49
image_mosaic_face_recognition.py	10....KiB	py File	2023-1...전 2:22
result.jpg	72....KiB	jpg File	2023-1...후 3:46
result_video.mp4	1.02 MiB	mp4 File	2023-1...후 3:49
video_mosaic_face_recognition.py	15....KiB	py File	2023-1...후 3:47

Directory: MyImage

Files of type: All Files (*)



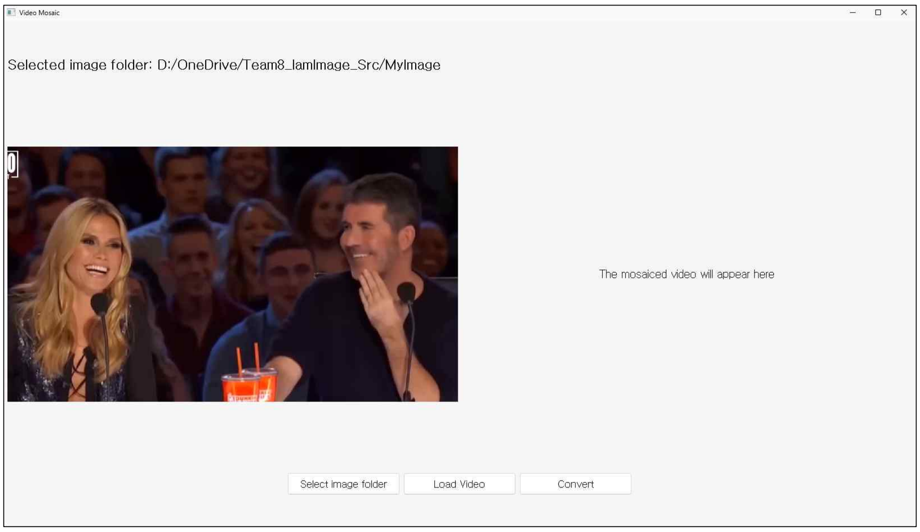
Selected image folder: D:/OneDrive/Team8_IamImage_Src/MyImage

The video to be

ear here

파일 이름(N): agt

Video Files (*.mp4 *.avi *.mkv)



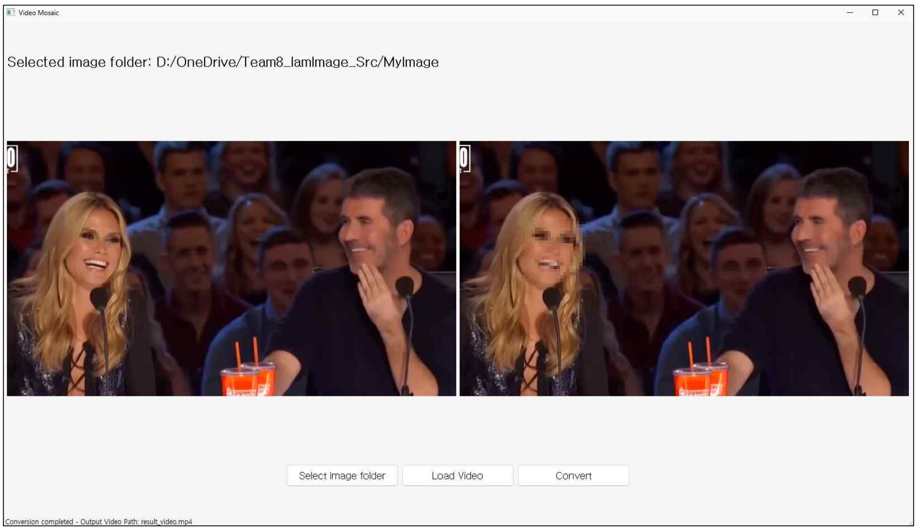
Selected image folder: D:/OneDrive/Team8_IamImage_Src/MyImage

The mosaiced video will appear here

(Video-3) Mosaic

- Start the video mosaic using the "Convert" button.
- After dividing the video into frame units and applying the photo mosaic technology to each frame, it is combined again to make a video.

  If the amount of calculation is too large, it can take too long, so the total amount of calculation was reduced by calculating only once in three frames.
- At this time, the people who are determined not to mosaic are the people who are included in the image in the previously created folder, and people other than those people are mosaiced.



Selected image folder: D:/OneDrive/Team8_IamImage_Src/MyImage

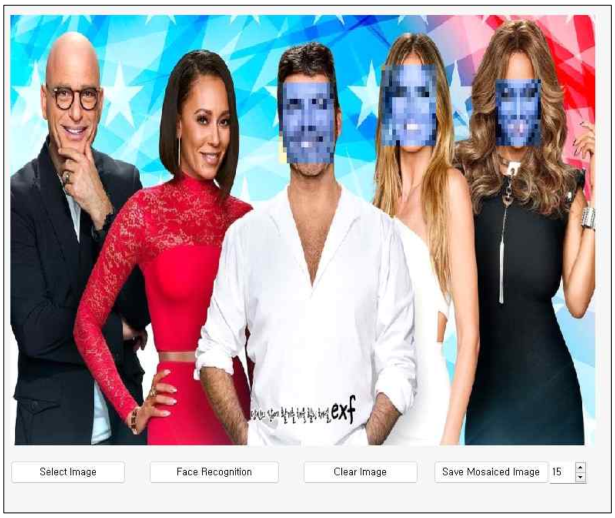
Conversion completed - Output Video Path: result_video.mp4

(Video-4) Save the video

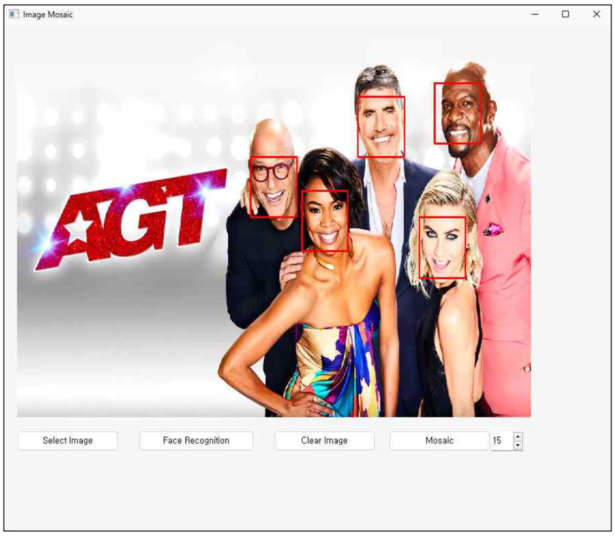
- The mosaic-completed video is automatically saved as "result_video.mp4“ in that folder.

(Add) Correct the error

- During photo mosaic, there was an error that the mosaic part appeared in blue. When I looked it up, it was a problem that occurred because the picture was expressed in BGR, not RGB. So, manually changing the red and blue values solved the problem.



- In the picture, the face area is recognized and displayed as a red box, but there was a problem that the box could not be clicked properly because there was a part where the actual displayed red box and the click area were misaligned. So, when I changed it by applying the ratio of enlargement or reduction when displaying the picture on the x and y values determined when clicking, it was solved.



3. The project time-line

November 9 (Thursday)

- have an idea meeting
- ideas such as OCR application, face recognition for fashion recommendation, mosaic for SNS upload
- confirm the topic with mosaic for SNS upload
- plan a photo mosaic function and a video mosaic function
- share roles such as report, ppt, etc.

November 16 (Thursday)

- share coding roles
- desing program flow

November 19 (Sunday)

- implement mosaic function

November 23 (Thursday)

- implement photo mosaic
- find some errors in photo mosaic

November 30 (Thursday)

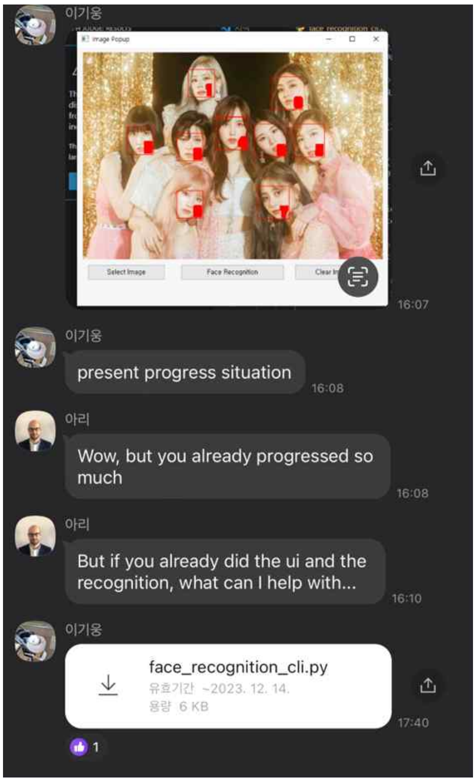
- resolve the errors in photo mosaic
- implement video mosaic

December 1st (Friday)

- complete tests

4. Design process of group discussion

Face-to-face meetings are held more often than non-face-to-face meetings, so there are few recorded data, but related data are attached.



이기웅

16:07

이기웅

present progress situation 16:08

아리

Wow, but you already progressed so much 16:08

아리

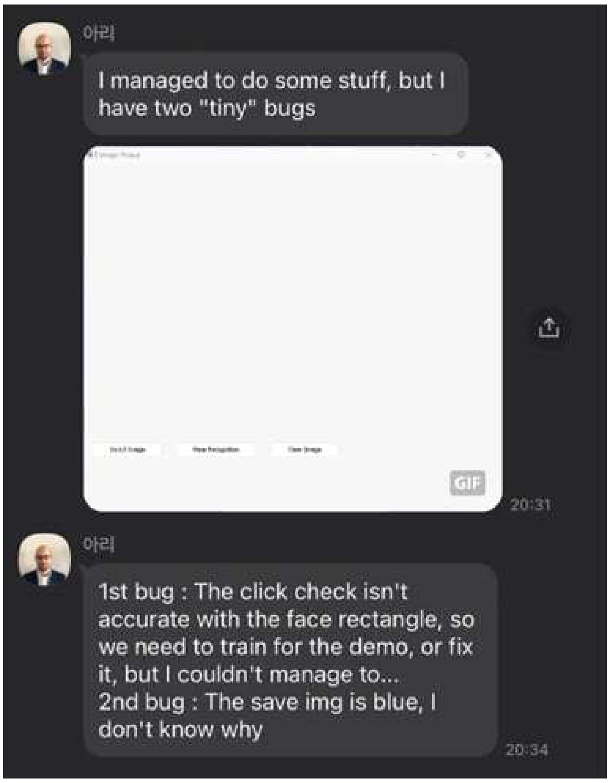
But if you already did the ui and the recognition, what can I help with... 16:10

이기웅

face_recognition_cli.py
유효기간 ~2023. 12. 14.
용량 6 KB

17:40

1



아리

I managed to do some stuff, but I have two "tiny" bugs

20:31

아리

1st bug : The click check isn't accurate with the face rectangle, so we need to train for the demo, or fix it, but I couldn't manage to...
2nd bug : The save img is blue, I don't know why

20:34

5. Team's thoughts on this project

20190402 Lee Jaemin

It wasn't easy to use machine learning while working on this project. Also, there were times when I was faced with linguistic limitations because I had never done a team project with a foreigner before. However, it was fun to combine the blur & mockery I learned in class with other fields and work on the project. Also, there was a lot to learn in the process of explaining the project and sharing opinions in front of the students as a team leader. Also, it was a meaningful time to understand and have fun with video processing.

20191127 Kim Doohong

Recognizing a specific person's face was the most memorable part of the technology. Finding only a specific person's face was much more difficult than recognizing all human faces, which was very complicated because it was a method of finding the area corresponding to the face in the picture and extracting and comparing features such as eyes, nose, mouth, and face shape from that area. And the difficulty was high because a lot of shape transformation or size adjustment occurred to process the technique in image mosaic processing, but when completed, the sense of achievement was high.

This project has high practicality from the idea, so I thought that if we make it well, many people might actually use it. Among the skills I learned in video processing during the semester, I'm satisfied that I made a program that can be used in practice by using mosaic with AI. It was my first time working on a project with a foreigner, but I was able to complete a good program well because everyone worked hard. This time, we showed the function with a simple UI, but if we have time in the future, I think it would be good to make it an app and make it available to others.

20194921 Lee Giwoong

In order to solve the face color problem that occurred while using the Face Recognition library, we analyzed the library's operation principle and investigated what color space it uses. The reason was that after face recognition, post-processing was processed in bgr order rather than rgb order. As a result, we found the cause of the problem with color conversion and corrected the code to solve the color problem of the face recognition result. Thanks to this, we realized that we need to continue to learn and improve when facing problems.

In addition, I was able to use the theory in practice by applying the mosaic technique I learned in class to the project. I thought about how the skills I learned in class could be applied to the project. It was a time to think about what new value could be created when combined with existing technologies with what I learned in class, and it enriched the learning experience.

In the process of designing and deploying the UI with PyQT, we designed the layout in consideration of user experience and convenience. Initially, the design was not perfect, but we prioritized programs that worked well. We came to the recognition that function is key, and that users should be able to understand and use it easily. Since it is always possible to improve the design later on, we initially felt that it was good to focus on the functional aspect. This approach will allow users to use the program well, and we think that we can provide better convenience to users through future design improvements.

50231583 Ali Mhrez

The project was nice, initial idea was great. Teamwork and task repartition were ok, maybe we could've organize a bit better by creating a github repository, and a requirement.txt / .toml to install the dependencies. Final result is great, the UI could be improved, but since the main functionnality works perfectly, I believe it is nice.

6. Etc. - Divide up

20190402 Lee Jaemin

- Team leader
- Draw up planning report & presentate the planning
- Implement the mosaic function
- Help to write final report

20191127 Kim Doohong

- Implement the video mosaic
- Help to implement mosaic function & photo mosaic
- Correct the error in photo mosaic
- Draw up final report

20194921 Lee Giwoong

- Implement the whole UI & photo mosaic
- Make planning PPT & final PPT
- Help to implement video mosaic
- Help to correct the error in photo mosaic

50231583 Ali Mhrez

- Implement the photo mosaic
- Draft the presentation script
- Help to draw up planning report & final report