Project: Muggle Magic

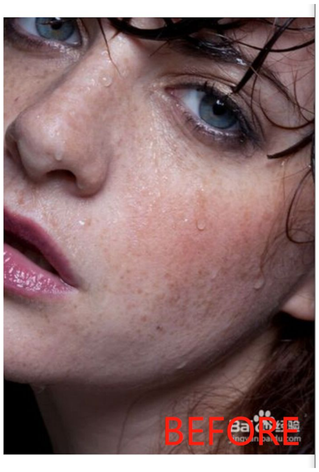
Members: Muyu Deng, Shanli Ding

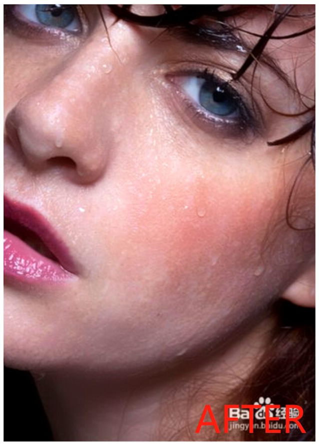
Initial idea and goals:

- Firstly, we considered making a web-app for searching similar photographs or images within the whole internet, like google's photo search with users' given picture. However, there are lots of existing implementations for this project, either on python or on MATLAB. Thus, we switched our project to photo processing, our project Muggle Magic. Besides, our initial goals goes from choosing which part of photo processing we want to do to choosing which platform we need to use. Finally, we decided to use Python as our script language, and two different image processing functions.

Applied algorithms and hypothesis:

- Within our project, there are two image processing functions which are younger magic and pencil magic. The main color blending algorithm we use is based on Photoshop documentation and utilized by some Photoshop problems posted by other users.
- As for our first Younger magic function, we have applied the image mixing algorithm on the original image. Specifically, we try to imitate the same performance on what happens on Photoshop. We have three different copies of the original image, and lastly mixing them together. Moreover, as for our Pencil magic function, we find a fantastic paper which is published by Hongkong University[1]. Because we have learned to read other people's papers and then implement the algorithm throughout the semester, we decide to challenge ourselves to try to implement this interesting feature. Now, we have finished up scripting our source code for both magic functions and as well as our websites' scripts. And all functions can work properly, but we did not link the backend and frontend together. However, the linking task is not impossible, so we believe that we will link them together and try to post on a website. And then let our friends enjoy the magic.





[1] Lu C, Xu L, Jia J. Combining sketch and tone for pencil drawing production[C]

Data Analysis and results:

- The main data analysis we did is analyzing how colors combine together, because there are different types of color in python, like RGB, BGR, L, etc. Plus, each color types' transformation is not simply like RGB switches orders then we get BGR. In our magic project, we have applied our own color mixing algorithm on final output images. Because sometimes it is easier and better to use BGR type and sometimes it is easier to use RGB, each type of color can help us do different color mixing.

Computer vision methods:

- First of all, we have applied filtering for our younger magic function. And secondly, we have learned and applied histogram matching and alpha blending on images. More importantly, we try to vectorize pixel processing as much as possible and avoid nested loops.

TODO:

- There are two things we still need to think about, which are run time for face detection by using CNN and linking the frontend and backend. The first problem is that we did experiments on face recognition on the internet, and its default algorithm's accuracy is too low to use. However, there is an option of CNN, and the accuracy is totally boosted, but the running time will cost two times more than the default one.

Known limitations and possible extension:

- We concentrate on this computer vision project, so we want to finish it up to 99% of completeness. Until now, the most important limitation is that our project cannot receive a photograph which contains lots of faces ,like family photos, then users will need to wait approximately 3 minutes to finish detection. Plus, our program is subjected to the complexity of image's color distribution. More detaily, if a user provides an image with plenty of colors like a snapshot with a garden, our program will need more time on processing convolutional layers. Thus, users need to wait for a little bit. And one more critical limitation is that all functions have fixed parameters for image processing, so there exists a problem that our function may not perform well on users' input images. Besides, deducting the running time on face detection will be our first future task. Moreover, as for future extensions, we have decided to add more features for this project during summer. New features may be cartoon-izing which is different from pencil sketching, cyberpunk filter which can adjust the saturation and cold and warm color for the whole image and (maybe) long leg filter which functions for making people's legs slim and long in the right ratio.

Contributions:

- Apprentice Shanli:
    - I mainly did the backend scripting. And I am a script tester as well, assures scripts run properly, because Muyu will take part in limitation testing. I searched for pencil sketching's paper, and discussed it with Muyu. And I also wrote the program report and explanation for Muyu, because both of us need to know all about our magic processes.
- Apprentice Muyu:
    - I contributed to the frontend development. And I am a limitation tester that I survey with how users will use our functions and what output format they may need. And also including finding the program's potential problems and discussing them with Shanli. My part is mainly on testing and analysis. Plus, considering users' experience with our project. Plus, I discussed with Shanli about all processes for our project, and we agreed that I can do the project demo presentation like introducing a product to real users.

Things we learned:

- As we go through the entire project, we have learned lots of things which are not only new algorithms and technologies but also new perspectives for computer vision. Firstly, we learned how to translate MATLAB to Python. Specifically speaking, we have done some experimental tests on MATLAB at first due to the existing tool box. And then we need to do the same thing in Python, so we try to translate content from MATLAB to Python. Secondly, we have learned how HDR works. The reason we learned HDR is because we need to know what histogram matching does within images. Thirdly, it is really helpful to learn other people's new algorithms and implement them by hand. Published paper contains plentiful details on describing how this algorithm works and how to apply mathematical equations on it. Thus, we think that learning how to read and analyze paper is also a big step we made. By the way, it is really magic that when we mix different layers of image together, we will get a fancy output.

Advices for future students:

- Algorithms for vision can really bring students with great satisfaction. Instead of other algorithms which are boring and always dealing with a bunch of data, we can see the picture's change straight forward, it really looks like magic!
- Students who would like to take this course really need to have strong linear algebra knowledge or it will become pretty tough then.
- Computer vision class is not an easy class, and it is totally different from computer system classes. Therefore, everyone needs to be prepared for the vision class. Both of

us are undergraduate, and we also recommend other undergraduates to take this class. Vision class will bring you into another world of computer science which is not like purely data analysis. And future students would better have interests and patience for this class, because both of us always get stuck into some problems which cannot find exactly the same problem on google. Therefore, vision questions need us to play around with other related tools and functions, and then we can get closer to the question's answer. Lastly, computer vision class is not easy, but we can assure that it will give you the sense of fulfillment for your progress.