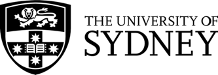
IMAGE COLOR STYLE MIMICKING FILTER APPLICATION

**Project Proposal Report**

****

**COMP5405 Digital Media Computing**

**Group Members**

mhua8500 (470141560)

yuli2840 (470005815)

hliu2547 (500185933)

## Application

### Background

As the rapid development of Internet transmission technology, increasingly more people start to convey information by images and video instead of only using text. Furthermore, the emergence and rise of social media applications (such as Instagram and Facebook) dramatically increased the people’s demand for image processing. Increasingly more people process their photo by various filters before posting them on social media to show their art taste and their pursuit of fashion. In this context, a large number of image processing tools that try to satisfy young people's pursuit of individuality are produced, released, and compete with each other. However, current applications such as Instagram only support limited filters which may not satisfy the users’ image processing requirement for those who want to show their unique personality and lifestyle. According to current mobile applications, it is not convenient for users to customize their own filter. Furthermore, users may be confused by the name of the downloadable filter and cannot easily find the suitable filter they want which leads to waste a lot of time on filter choosing.

This proposal will describe an image color style mimicking filter application. This application can build a special filter based on the style of a given image, and then apply the filter to a target image in order to transfer the target image to the same style with the given image. Also, this application should be light enough and easy to use on smartphones.

### Related work

A technology called image style transfer (Gatys, Ecker & Bethge, 2016) is published in the image processing domain. The Image style transfer technology can extract the artistic style from artistic printings and recombine the style with the content to a given natural image by a specific Convolutional Neural Networks (CNN) Algorithm. For example, the image style transfer technology can transfer your own picture to the same style of Vincent Willem van Gogh The Starry Night. In the same year, a feed-forward network is trained for image style transfer (Johnson, Alexandre & Li, 2016), the new algorithm is hundreds of times faster than Gatys’ method. This algorithm is now open sourced on GitHub. Furthermore, another open-source project called Deep Photo Style Transfer (Luan, Paris, Shechtman & Bala, 2017) which implements Gatys’ image style transfer technology.

The image style transfer technology is mentioned here because what it can do is similar with the goal of the application in our project. However, there are several differences between the image style transfer technology and our project (the Image style mimicking filter). The Image style mimicking filter application is focusing on extracting the “color style” of the given image, such as the hue, saturability, and value. Then, create a filter by these data, and apply the filter to the target image to transfer the style. The Neural Network algorithm is not in the scope of this project.

On the other hand, another Image Color Transfer technology (Reinhard, Ashikhmin, Gooch & Shirley, 2001) based on the lαβ color space (Ruderman, Cronin & Chiao, 1998) is doing the similar function. However, our team will choose another simpler method in the project rather than Reinhard’s method, due to the potential cost and complexity of implementing Reinhard’s method.

### Significance and competitive power

The image style transfer method given by Gatys is based on the Convolutional Neural Networks (CNN) and deep learning. However, deep learning requires a long processing time, even though Justin’s algorithm makes it significantly faster than the original one, it is still not fast enough for daily use. Clients may not want to wait for the long processing time, especially when they are trying to compare with different styles. Also, image distortion may be caused during the processing period without suitably chosen (lee & Tseng, 2019). Furthermore, the algorithm generated by deep learning is a black box, so it is difficult for the developers to change the detail of the processed pictures by adjusting the algorithm.

As a result, we are trying to implement the Image style transfer without the CNN in order to decrease the processing time and make the application more flexible and easier to use. Thus, people can liberate their creativity by producing a series of pictures with the same style or creating their unique filter for further use. This application could enrich people’s lives and help people to better show themselves on social media.

### Requirements

#### Functional requirements:

This application can build a special image color filter based by the color style of an input image. Then, the application will allow the user to save the highly individualized filter and use the filter to process other pictures.

#### Non-functional requirements:

This application is designed to allow clients to have a comfortable, fast, convenient and fun image processing experience. It aimed to help users show their unique art taste and pursuit of fashion.

#### User stories:

As a social media user, with the application, I can easily get a highly customised and individualized filter to process my photos before I post them to social media, so that I can show my unique art taste better.

As a photographer, with the application, I can build a color filter based on the color in the photo I choose, so that I can better show the emotion based by the color I choose in my photo.

As a smartphone user, with the application, I can easily turn the color of my photo without knowing complex input arguments and too technical terms, so that I can easily make my photo look better.

As an artist or graphic designer, with the application, I can change the color of my image easily and fast, so that my work efficiency may be increased.

## Solution:

In this section, we will explain the preliminary solution we designed to achieve the image style transfer function. However, the specific techniques we will use in the final application would be different based on the further research and development of our project. We divide our solution into two parts, which are the information extraction from the original image and the filter application to the target image.

### Information extraction

The first part mainly includes the different types of information we will choose to extract and the processing methods we will use to process the information we extracted from the original image. Since our style transfer application mainly aims to transfer the color style of an input image to a target image, we need to consider carefully about the color information we need to extract from the original image. We decided to use the K-means methodology to cluster the color values of each pixel of the original image, and then produce a list of K color values.

K-means methodology is a basic machine learning algorithm to clustering data, which will cluster the training data into K different partitions. In the application, we will use the K-means algorithm to cluster the different pixels and calculate the K centroids which are presented as RGB color values, and then use the K centroids as the main K colors of the original image. For the detailed procedures of this step, we will first reshape the original image from a matrix with three RGB channels which is [p, q, 3] to a list with RGB channels which is [p\*q, 3]. Next, we randomly select K centroids as the initial centre value for each K cluster. Then, we will calculate the Euclidean distance for each pixel to each cluster centroid, and determine the nearest clusters which each pixel belongs to based on the Euclidean distance. In this case, the Euclidean distance means the similarity between two color pixels based on their RGB values. After one iteration, we will recalculate the centroid of each cluster based on the mean value, so that we can get a new centroid for each cluster and calculate the Euclidean distance again. After a fixed number of iterations, we can find the clusters of each pixel it belongs to and determine the final K centroids value. In this part, there are 2 parameters that need to be fine-tuned which are K value and iteration number. If the K value is too big, it will be more difficult to apply the color filters in the next step. If the K value is too small, the colors could not represent the original style completely and we will lose many details. Also, if the iteration number is too high, the time taken for the first step would be extremely increased. On the other hand, if the iteration number is too low, we may get a bad clustering result which cannot represent the real color style of the whole original image. Thus, in the development step, we need to carefully fine-tune and test the K value and iteration number and find the best value for them. The final K centroid value should be the main K color values we extract from the original image, and we will build a filter based on those K colors to apply to the target image in the next part.

### Filter application

The second part would be the method of how we will modify the target image using the color data we extracted from the original image. In this step, we have two options to achieve this function.

The first one would be using the applyColorMap() function in the opencv package to apply the colors. First, we will create a color map based on the K centroid colors we extracted from the original image in the first part. Since a colormap in python is actually a gradient color list represented in hexadecimal values. Thus, the algorithm will transfer the RGB decimal colors in the k centroid colors to hexadecimal colors, and create the colormap. The colormap will be presented as a gradient color bar using the K centroid colors. Then we will use the applyColorMap() function to apply the colormap we created to the target image. This method is the most convenient one and easy to develop. However, the applyColorMap() function has the limitations of preventing the original color style for the target image, since the function will treat the image as a grayscale image to process. Thus, we have another option to achieve the filter function.

The second algorithm will first transform the K centroid colors from RGB color to HSV color, so that we can make use of the hue value. Then, we need to set a range for the hue value of K centroid color such as +/- 20. Next, we will scan the target image and transform the RGB value to HSV value for each pixel. If the hue value for each pixel of the target image falls in one of the range of K centroid colors, we will set the hue value as the nearest K centroid value without changing the saturation and value. In order to balance both the color style of the input image and the target image, we need to carefully decide the range of the hue value we want to change. The algorithm will set the similar colors in the target image as their similar K centroid color in the input image, so that we can get a similar color style of the input image for the target image, and at the same time, it will retain the original color style to the greatest extent for the target image. Also, we need to find the most appropriate range value in this algorithm.

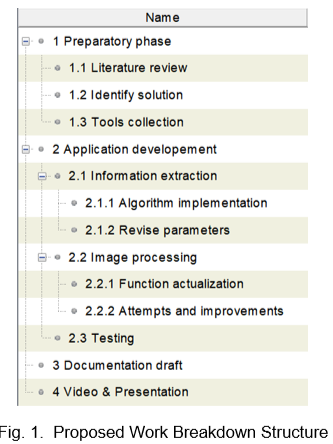
The final result after the two steps would be the edited target image containing the color styles in the original image. The solution we explained in the proposal is an initial plan for our development. There are some parameters we need to fine-tune. Also, the current solution needs to be modified in the development step, and some of the detailed functions and algorithms in the solution also need to be tested through the real database. The two solutions in part two need to be compared and we have to choose the algorithm which achieves the tasks most perfectly. We will spend the following weeks to test the feasibility and improve the algorithms for our image style transfer application.

## Plan and Schedule

Based on the designed solutions, the project can be divided into three sections: preparatory phase, application development, documentation draft and presentation make. In each section, one task will be implemented in multiple deliverables ensuring the whole project under control.

### Individual tasks

In order to concisely exhibit the tasks required to be solved, Work Breakdown Structure (WBS) is utilized to describe our future plan. “Fig. 1” is the WBS organized in tabular form.



The task of preparatory phase refers to the preparation before application development, and there are three deliverables within the task:

1) Literature review: By reviewing more materials with resemble topic, improve the details of original project plan, so as to increase the feasibility and novelty of the solution;

2) Identify solution: After learning more about literature, we will modify the solution once needed to meet the expectations. During the period, the solution will be finalized;

3) Related tools: We are supposed to prepare related tools for projects in this section, such as development environment, references and programming skills.

Then, we will start programming the application following the instruction of the solution. At this stage, in order to finish the task of delivering features of one image to another and actualize the solution, there are two key parts required to be completed in phases: the information extraction on source image, and the filter processing of target image. Furthermore, we have more detailed tasks in each phase:

1) Algorithm implementation: Based on the designed solution in proposal and further learning, we will utilize K-means methodology to obtain parameters that are used in image processing. At this stage, the expected function of the clustering and extraction algorithm is supposed to work smoothly;

2) Revise parameters: As mentioned above, the number of iterations and K value are two main parameters that need to be adjusted well. This task requires us to figure out most appropriate K value which is the number of colors that will be extracted from source image, and the number of iterations which ensures that those K colors can represent the style of image;

3) Function actualization: With the help of two parameters obtained in the extraction phase, construct the algorithm and apply it to the target image, to get a result of similar style. At this stage, we will try both solutions to figure out the better one and regard it as the final solution;

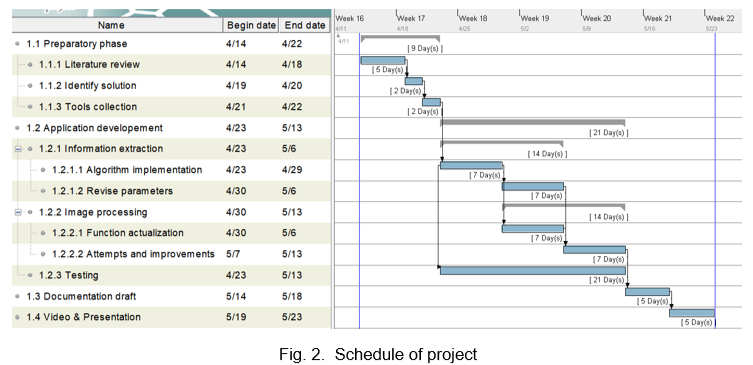
4) Attempts and improvement: Use the application to implement varied images and improve the program, such as a wider recognition of source image or more choices in operations, to make improvement of application;

5) Testing: The testing will keep lasting during the development phase, to compare two possible solutions and select the one with better effect to use.

During the development, we will always test the function and program by applying techniques to find errors and make improvement (Ehmer Khan, 2011). After finishing these tasks, draft the required documentations, and make the introduction video and presentation.

### Schedule

According to the WBS and estimated effort we need to put in each task, we design the schedule in the Gantt Chart of the whole project in “Fig. 2”, which can be used to monitor the project progress and make dynamic adjustments.



The project is expected to finish in 40 days, and the buffer time is taken into the consideration of durations.

## Reflection on proposal writing

To draft a clear and concise proposal, we have consulted many related works and articles to gain necessary resources. For example, the knowledge of how to write literature reviews (Ehmer Khan, 2011) and the study method of digital media computing application (Criminisi, Perez & Toyama, 2004) . With the help of these materials, we draw on helpful documents and integrate our research methods.

Besides, according to some books describing proposal writing (Forsyth, n.d.), we categorize the proposal into different sections, to demonstrate the formation and change of our ideas with a gradually increasing feasibility. To come up with a reliable solution, we also look through the article with the same topic (Reinhard, Adhikhmin, Gooch & Shirley, 2001) to generate our proposal.

What’s more, to extend a schedule to monitor the progress, we learn the skills of creating a Work Breakdown Structure (Indelicato, 2009) like dividing tasks into deliverables to finish them step by step, and Gantt Chart (s.r.o, 2021) representing the sequence of each task to ensure the proposed plan is feasible and reasonable.

Overall, by referring to a wide range of resources, we generate our application idea, project plan and this proposal. In the final phase of the project, we will review more literature to keep improving our solution and documentation draft.

## References

Criminisi, A., Perez, P., & Toyama, K. (2004). Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transactions On Image Processing*, *13*(9), 1200-1212. doi: 10.1109/tip.2004.833105

Ehmer Khan, M. (2011). Different Approaches To Black box Testing Technique For Finding Errors. *International Journal Of Software Engineering & Applications*, *2*(4), 31-40. doi: 10.5121/ijsea.2011.2404

Forsyth, P. *How to write reports and proposals*.

Gatys, L. A. Ecker, A. S. & Bethge, M. (2016) "Image Style Transfer Using Convolutional Neural Networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2414-2423, doi: 10.1109/CVPR.2016.265. Retrieved from: https://ieeexplore-ieee-org.ezproxy.library.sydney.edu.au/document/7780634

Indelicato, G. (2009). Book Review: Building a Project Work Breakdown Structure: Visualizing Objectives, Deliverables, Activities, and Schedules. *Project Management Journal*, *40*(3), 85-85. doi: 10.1002/pmj.20133

Johnson, J. Alahi, A. & Li, F. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. Retrieved from: https://cs.stanford.edu/people/jcjohns/eccv16/

Lee, P., & Tseng, C. (2019). On the Layer Choice of the Image Style Transfer Using Convolutional Neural Networks. *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 1-2.

Luan, F. and Paris, S. and Shechtman, E. and Bala, K. (2017).Deep photo style transfer.Retrieved from:

https://github.com/luanfujun/deep-photo-styletransfer

Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. *IEEE Computer Graphics And Applications*, *21*(4), 34-41. doi: 10.1109/38.946629

Ruderman, D., Cronin, T., & Chiao, C. (1998). Statistics of cone responses to natural images: implications for visual coding. *Journal Of The Optical Society Of America A*, *15*(8), 2036. doi: 10.1364/josaa.15.002036

s.r.o, B. (2021). GanttProject: free project management tool for Windows, macOS and Linux. Retrieved 10 April 2021, from https://www.ganttproject.biz/

Webster. (n.d.). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly : Management Information Systems., 26(2), 13–23. https://doi.org/info:doi/