COL783: Digital Image Analysis

Assignment 1: Artistic Image Enhancement, and Style Transfer

Team Members: Aditya 2021CE10494, Atharv Saxena 2021ME21055

Assumptions

The assumptions are :

- No noise reduction is performed on input images.
- Relatively Simple images will be used for artistic style transfer.

Part 1: Simple Image Enhancement

1.1 Taking input

Firstly we have read the image using the cv2 inbuilt function cv2.imread(), then we have converted the image from BGR to RGB as cv2 reads in BGR format. Then the image is normalized

1.2 Shadow Map Generation

RGB to HSI: We have converted the RGB image to HSI space using the matrix operations given in the writeup. There are little differences between the HSI generated by the cv2 function and our function due to different methods and inversion of the S channel. After obtaining the HSI image, we normalize it.
r-map generation: r-map is generated using the instruction given in the writeup and then it is scaled for thresholding
Thresholding: We have done the thresholding as instructed in the writeup, we have took inspiration from otsu's thresholding as the methid provided in the writeup was similar to it. We generated the **shadow map** after thresholding.
Shadowed image: We generate the shadowed image using the shadow map and we took $\lambda = 0.6$ after testing multiple values.

Input Image

Shadow map

Figure 1

1.3 Line Draft Generation

Bilateral filtering: We apply bilateral filtering to the greyscale converted orignal input image. The code is explained via comments in the file itself. The chosen values for the parameters are $diameter = 7$, $\sigma_{color} = 20$, $\sigma_{space} = 20$
After experimentation with the parameters, we found out that increasing the diameter improves the quality of the filtering but increases the computation cost and decreasing the diameter gives a poor result.
Edge map: We create the edge map by calling the detect edges function defined in our code.

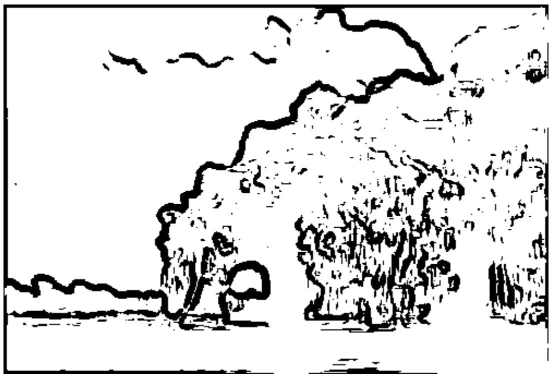
edge map 1

edge map 2

Figure 2: example edge maps

Here I present some sample edge maps generated.

Thresholding and Line-draft generation: We have taken threshold to be 0.3 after experimenting with different values.



Line draft 1

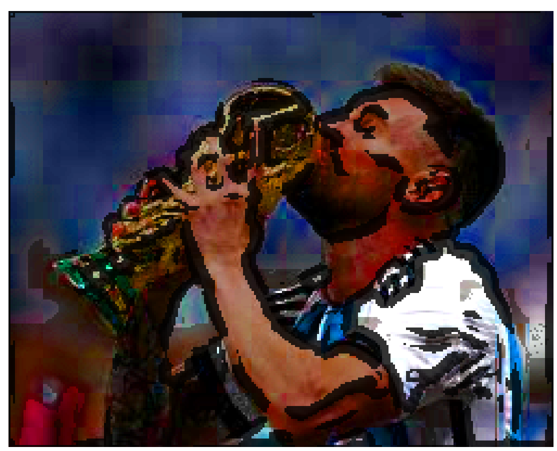
Line draft 2

Figure 3: example line drafts

1.4 Color Adjustment Step

Shadow image enhancement : We have enhanced the shadow map using the steps provided in the report and we have chosen $\rho = 0.2$ after experimenting. And then we have performed saturation correction by scaling the S channel by a factor of 2.5 after converting the image to HSI.
Finally, We obtain the **Artistic Rendered Image** and we chose $\beta = 0.2$ for the formula provided in the writeup. Here are the results.

Artistic Rendered 1



Artistic Rendered 2

Figure 4

Part 2: Quantized Rendering

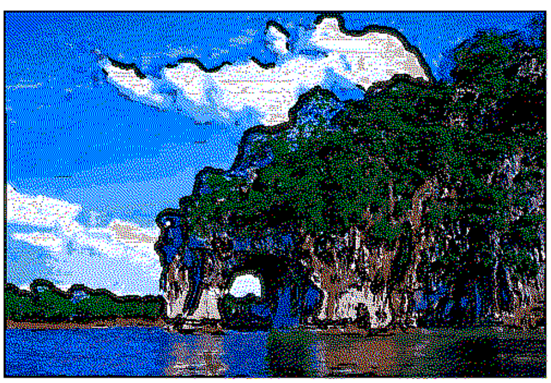
Median Filter: We have used the quantized the image to 2^n colors using the median cut algorithm. In these examples we have taken the n (depth) to be 3 and 5 for different images, thus reducing the image to 8 and 32 colors respectively.

median cut with depth 5

median cut with depth 3

Figure 5: images after median cut

Floyd Steinberg Dithering: We have applied Floyd Steinberg Dithering to the quantized image, to produce a more smoother transitions as perceived by the eye.



Floyd Steinberg with depth 5

Floyd Steinberg with depth 3

Figure 6: images after median cut

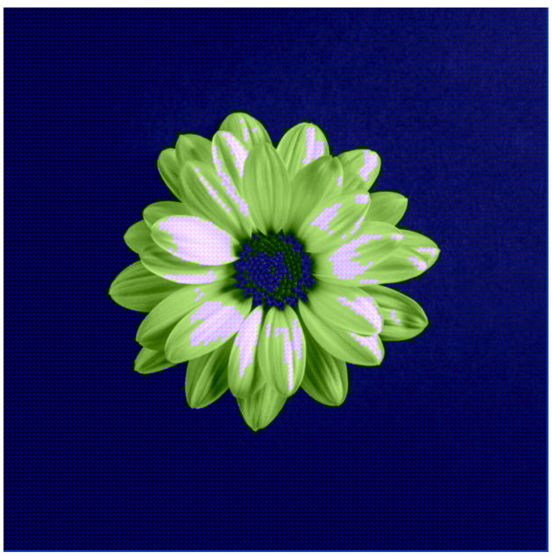
Part 3: Artistic style transfer

We have done the artistic style transfer by manually defining swatches in the image. We have picked relatively simpler images for the demonstration so that we can run the code for less number of swatches. We have observed that the style with Floyd Steinberg dithering provides better results compared to the style without Floyd Steinberg dithering applied. Here are the results for a simple flower image.

median cut

with floyd

Figure 7: reference images



median cut

with floyd

Figure 8: style transferred images

Hyper-parameters :

part1:

hyper-parameters	**values**
λ(shadow image)	0.6
diameter(bilateral filter)	7
σ_{space}(bilateral filter)	20
σ_{color}(bilateral filter)	20
Threshold(line-draft)	0.3
ρ(shadow image enhancement)	0.2
Linear saturation scaling factor	2.5
β(artistic rendering)	0.3
Error coefficients	7/16, 3/16, 5/16, 1/16

Discussions :

- **Shadow map :** We can see in figure 1 that there is some noise (on the cloud) in the shadow map generated, this could be due to noise in the original input image or the rgb to hsi conversion, we observed differences in the hsi generated by the opencv standard library and our hsi function, this is due to the different algorithms and also the inverted S channel generated by our function.
- **Line draft :** Observing figure 3 we can see that some edges are not very clearly defined (the hair and right hand in line draft 2), this is due to the selection of our threshold. We wanted to eliminate extra details but that also lead to elimination of some necessary details like the boundaries of the main object. This is due to lower gradients in intensity in those regions.
- **Artistic rendered image :** After experimenting with the values of ρ and β we observed that lower values of ρ gives image with higher brightness and contrast. Lower values of β gives thick and well defined shadows and higher values of β gives very subtle shadows. Hence, we chose a low value of β as we wanted strong shadows. This also lead to some areas where the shadow is stronger than required (eye and ear of Messi, figure 3).
- **Median cut :** The algorithm works correctly and gives accurate results.
- **Floyd Steinberg :** After experimenting for multiple values for the error coefficients, we observed that we were getting the best results for error adjustement in 4 directions (different from what was given in the paper), with the selected error coefficients.
- **Artistic style transfer :** We fail to observe strong shadows in both images of figure 8, this is due to relatively simple choice of greyscale image, less number of swatches (i.e 2) and weaker edges in the reference image. But the algorithm was succesfully able to distinguish the flower from the background and colored it accordingly, it also retained the shadow lines on the petals.

Conclusion :

The code has performed well at least up to part 2, we have generated satisfactory results and were able to successfully make the input images look like a painting. A little bit more experimentation was needed from our side in part3. Taking swatches manually is very inefficient and is impractical for images with multiple features but it provides good results for simple images.

References

[1] author = Muthukrishnan, title = Reducing the number of colors of an image using Median Cut algorithm, month = October, year = 2019,
link =https://muthu.co/reducing-the-number-of-colors-of-an-image-using-median-cut-algorithm/

[2] title = Floyd-Steinberg Dithering,
link =https://www.visgraf.impa.br/Courses/ip00/proj/Dithering1/floyd_steinberg_dithering.html#:~:text=The%20error%20dispersion%20technique%20is,you%20have%20not%20visited%20yet