Image Cartooning

Project Id: 18

1 Team Members

- K R Prajwal (2019701002)
- Sindhu B Hegde (2019701021)

2 Github Link

https://github.com/prajwalkr/cartooning

3 Goals of the Project

The main goal of our project is to implement [1]. Given an image, we would like to produce the cartoon effect for the image using the cartooning algorithm. We seek to emulate a particular effect known as cel-shading in the computer graphics world. We would also analyze the results on various kinds of images such as portraits, indoor and outdoor scenes and try to improve the failure cases.

4 Problem Definition

Techniques in image processing can be applied to achieve aesthetically pleasing graphical effects on images. Cel-Shading is one such effect typically associated with the rendering process of 3D graphics. The effect is achieved by first quantizing the color shades so that the shadows cast on an object falling under continuous lighting will form discrete areas of color. Secondly, contour lines may be added to accentuate the outlines of the model. Our goal is to implement a version of the cel-shading effect for various kinds of images to achieve the cartooning effect as shown in Figure 1.

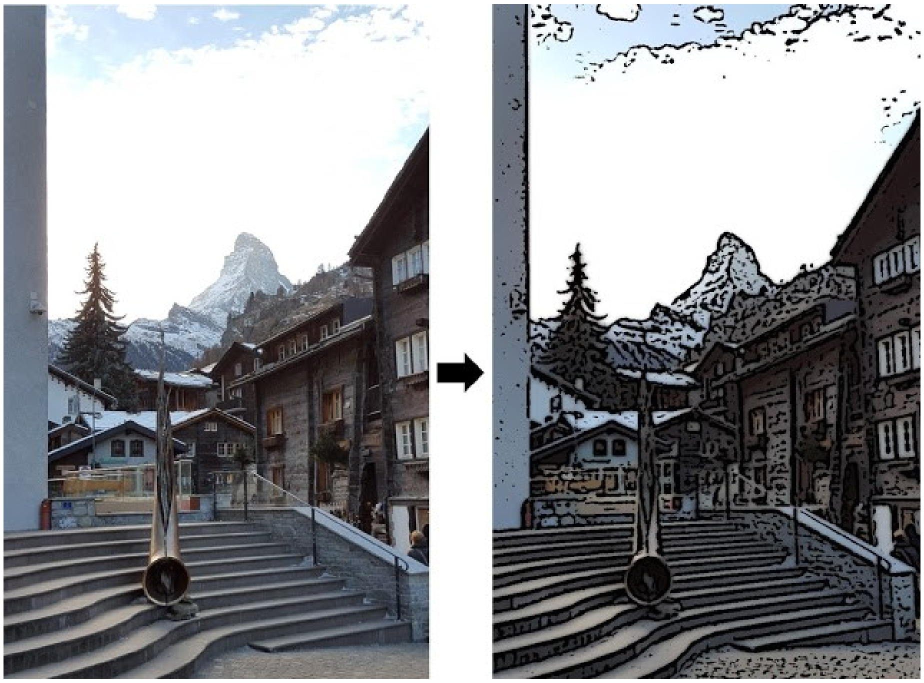

Figure 1: Expected Cartooning Result. Left: Input, Right: Output

The process to produce the cartoon effect is divided into two branches - one for detecting and boldening the edges, and another for smoothing and quantizing the colors in the image. Finally, the resulting images are combined to achieve the desired cartooning effect. The block diagram illustrating the steps of our approach is shown in Figure 2. In the subsequent sub-sections, we explain the steps we plan to take to achieve the aforementioned goal.

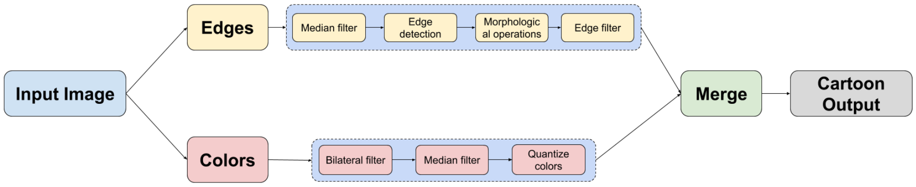

Figure 2: Block diagram of our approach

4.1 Edges

Finding smooth continuous contours is an important component to achieving the overall effect. The following steps are taken to provide contour detection that works in an artistically pleasing manner on a wide variety of images.

4.1.1 Median Filter

A median filter is applied to reduce any salt and pepper noise in the input image. We choose a small enough kernel size to preserve the edges.

4.1.2 Edge Detection

We plan to use the Canny edge detector present in OpenCV. The benefits of using the Canny edge detector instead of a Laplacian kernel is that the edges are all single pixel edges in the resulting image, which would be helpful to perform subsequent steps.

4.1.3 Morphological Operations

The purpose of this step is to both bolden and smooth the contours of the edges slightly. We plan to apply different combinations of morphological operations as the aesthetics of the contours in the final image depend largely on this step.

4.1.4 Edge Filter

Finally, the edge image is separated into it's constituent regions, and any region with an area below a certain threshold is removed. In this way, small contours picked up by the Canny edge detector are ignored in the final image, which helps reduce unwanted line clutter in the result.

4.2 Colors

The other important aspect of the cartoon effect is that of blockish color regions. In this branch of the algorithm, the colors are repeatedly smoothed to create homogeneous color regions. The colors in these regions are then re-quantized at a lower quantization.

4.2.1 Bilateral Filter

We apply bilateral filter repeatedly to smoothen regions and preserve edges. Since multiple applications of bilateral filter is computationally very expensive, we plan to down-sample the input image before applying the filter.

After applying the filter, we up-sample the image back to the original resolution.

4.2.2 Median Filter

A median filter is applied after restoring the image to its full size in order to smooth over any artifacts that occurred during the upsampling.

4.2.3 Quantize Colors

We apply the standard method to quantize images into a desired range. This is done to obtain uniform blobs of colors for the cartoon image.

4.3 Merge

Once both the color and edge stages are complete, the final stage is to overlay the edges onto the color image to produce the cartoon image output.

5 Expected Results

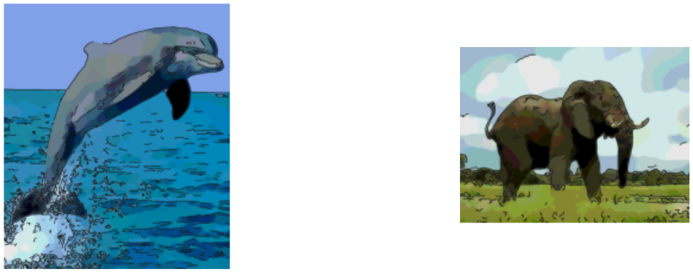

Figure 3: Final Expected results

6 Milestones & Expected Timeline

The tentative time-line for the project will be -

- Week 1 - Median filter and edge detection
- Week 2 - Morphological operations
- Week 3 - Edge filter
- Week 4 - Bilateral filter and median filter
- Week 5 - Quantize colors
- Week 6 - Merging

If time permits, we would also like to extend our approach to overcome the drawback of [1] to handle the shadows cast by the input images.

References

[1] Dade Kevin
Toonify: Cartoon Photo Effect Application
https://stacks.stanford.edu/file/druid:yt916dh6570/Dade_Toonify.pdf