TOONIFY

Aditya Sharma 170050043 Suraj 170050044 Rohan Abhishek 170050078

Abstract

We built a Toonify app that solves the purpose of converting a natural image to a "Toon" version of the image. The primary influence of the project was the following paper:

`https://stacks.stanford.edu/file/druid:yt916dh6570/Dade_Toonify.pdf`

Applications and Motivation

Toons are a trend today. Social media websites, newspapers, films, shows, advertisements, etc, all have different forms of cartoons. Therefore there is and there will be a need of softwares that deal with Toon editing, modification and construction. With this purpose in mind, we attempt to explore and discover various methods and approaches in the problem.

Techniques

The system is divided into three sub-categories:-

1. Canny Edge Detection along with Morphological Operations
2. Bilateral FIltering on CIE Lab Colour Space

1. Canny Edge Detection

Canny Edge Detection is chosen over other Kernels such as Laplacian is chosen because all edges will become single pixel edges. This allows Morphological Operators to act more predictably and in a desired way.

Canny is later followed by Morphological operation dilation which uses a 2x2 structure for dilation. This has 2 advantages:

- Reducing holes in canny edge and thus makes the contour smoother.
- The edge gets bolden and thus the cartoon looks more aesthetic.

2. Bilateral Filtering

Bilateral Filtering along with colour quantization is implemented for segmentation and clustering. We have typically used Bilateral filtering 30 times to toonify an image and 15 for a video(this can be increased).

However, the important point to note is that Bilateral is not applied to the input RGB image. Rather, it is first converted to CIE Lab colour space from RGB colour space, and then Bilateral Filtering is done. The advantage of CIE Lab space is that it is more oriented towards visual importance and hence is developed according to human perception.

The the step size for colour quantization is 8 in our case. Although the paper proposed that a quantization of 24 will be effective, we have observed avoidable artifacts and thus chosen the quantization to be a bit lesser.

Combining the results

Once we obtain the 2 parts of our final image, we superimpose them so that the edge gets aligned with the correct boundaries of the image.

Note: The threshold of canny edge detection plays a crucial role here. Whether the particular patch is highlighted in the image depends on whether that patch has highlighting edges. This is an advantage since we can tweak the parameters to get desired edges. But with this comes a disadvantage that we cannot control some other patches part of the image. **A Get-around:** A possible solution is to decide the parameters for the patch separately, but this is very effortful task.

Other Alternative approaches

The other approach would be to reverse the sequence of events. To elaborate, first we apply Bilateral filtering and on that output, we apply Canny edge detection.

1. Apply Bilateral Filtering using appropriate parameters.
2. Apply Canny edge detection on the output image of step 1.

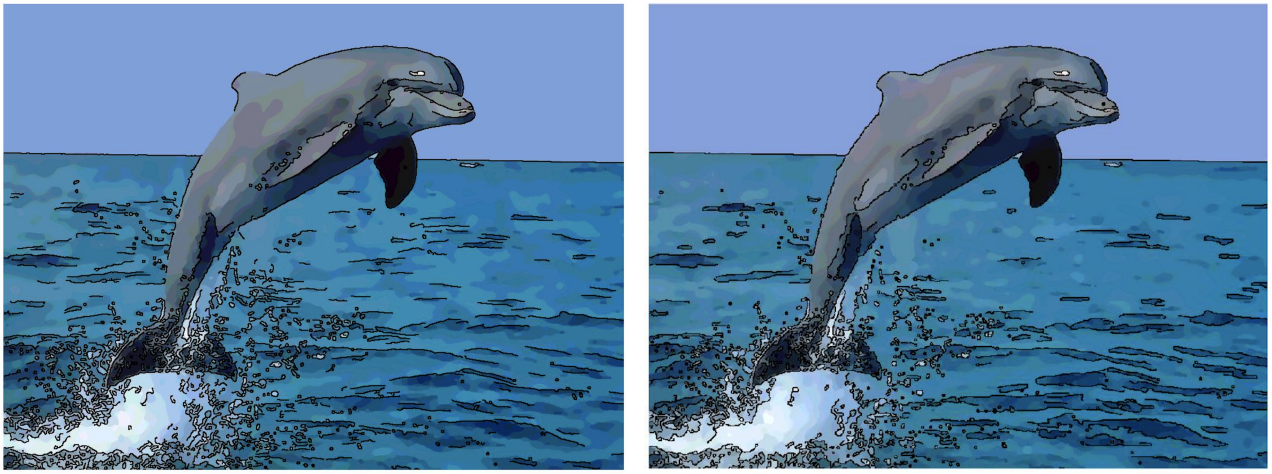

Figure 1: Separate canny and edge detection(left) vs Canny after bilateral filtering(right)

In Figure 1, we observe that we are at an advantage in doing Canny edges after Bilateral. You will notice some "good" boundaries appearing while some of the other boundaries in the sea region have disappeared. However, this is not always true as you can not in Figure 2.

Figure 2: Separate canny and edge detection(left) vs Canny after bilateral filtering(right)

We see a disadvantage of using Canny after bilateral filtering is that we may get some boundaries after the segmentation which we would not have got if the patch was smooth(like before segmentation). **A Get-around:** A possible solution is to increase the threshold value of Canny edge detection. While doing this, we may miss some good edges.

Experimental Analysis

We briefly discuss the experimental outcomes as well as situations encountered in the project implementation. We also discuss where the model needs to change parameters according to images.

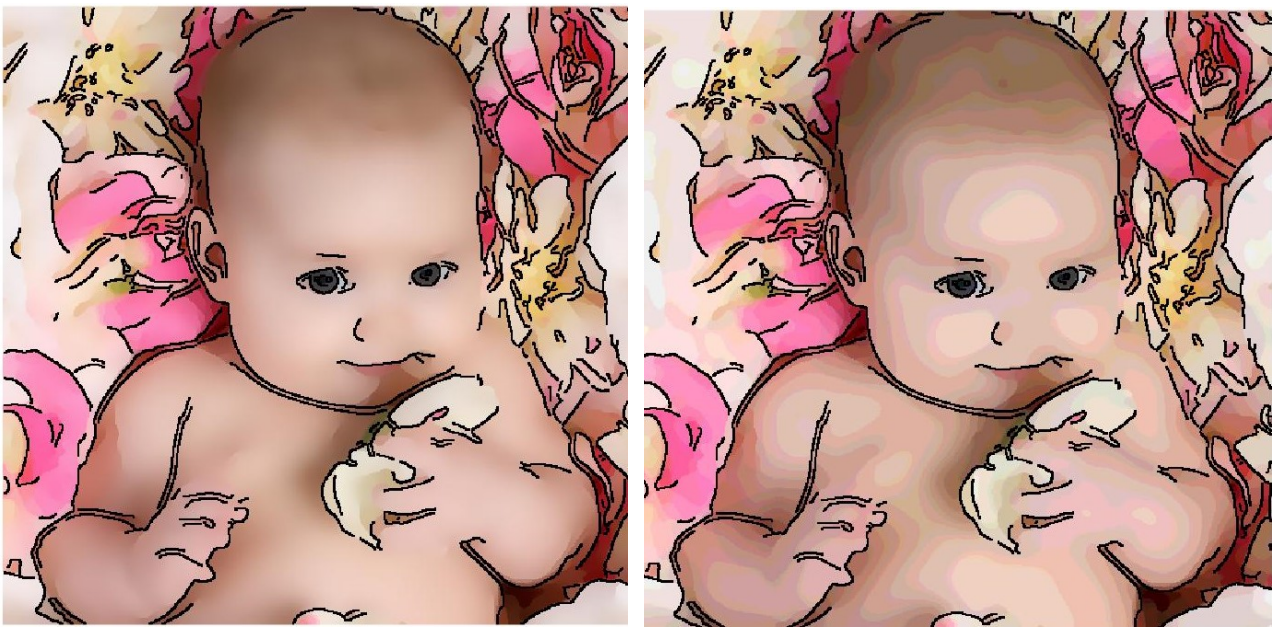

Figure 3: No quantization(left) vs Colour quantization(right)

Problem of Quantization: Figure 3 demonstrates the issue that certain images are ruined because of quantization. Quantization causes staircase artifacts which are undesired in certain images. **A Possible Solution:** We manually switch between quantizing and non-quantizing versions of the code to ensure better outputs.

Problem of Edge Thresholding: Tweaking parameters of edge thresholding impacts the image. This is demonstrated in Figure 4.**A Possible Solution:** We manually set the parameters to obtain the best image.

In the end, it is an interplay of these 2 factors which make the "Toonified" version of the image more desirable.

Challenges in Implementation

The following were some challenges while implementing and executing the proposed project:

1. Changing parameters for Bilateral filtering: This was a challenge since we had to find a spacial σ, degree of smoothing and quantization.

2. In canny edge detection, we weren't getting results at par out expectation. So we implemented our own canny edge detection and compared output(As shown in Figure 5). For uniformity, we will still be using MATLAB's function.

Figure 4: Effect of reducing Canny Edge Threshold

3. Dealing with videos toonifying: We need to find the optimal parameters which work on most of the frames in the video. Then we will also have to deal with some noisy frames, which we elaborate in proposed extension.

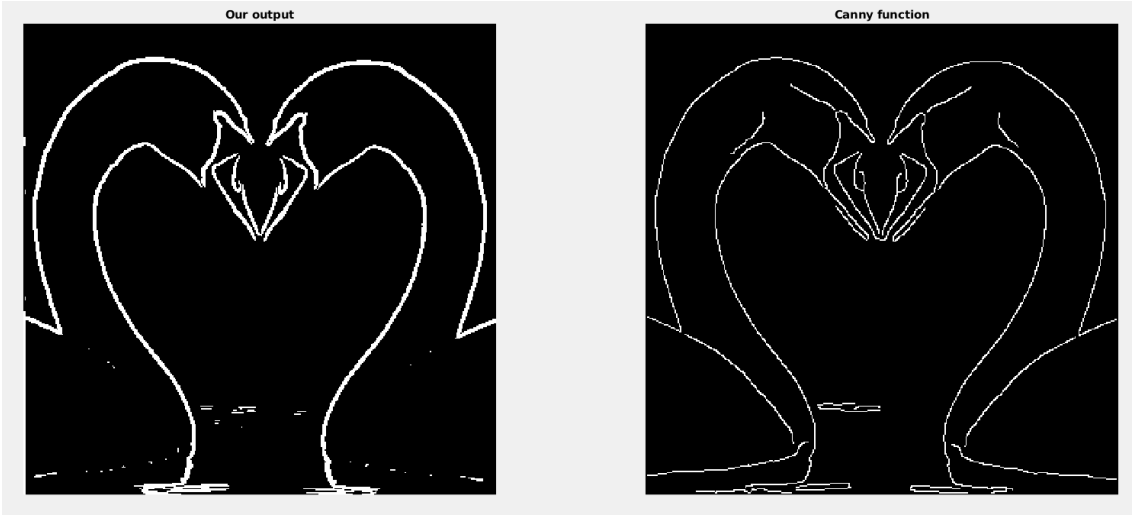

Figure 5: Canny Edge Detection after Dilation: Our implementation(Left) vs MATLAB Implementation(Right)

Proposed Extension

We have also extended this to videos. The following are figures that demonstrate the "Toonification of video and the frame by frame change in the toonified video encountered. We observe sudden shifts of edges in consecutive frames. This could be due to following reasons:

1. Change in actual frame intensities across the frames.
2. Error that is due to a particular noisy frame between good frames.
3. Noisy frames throughout the video and the error is not caused by a few frames.

While Case 1 is a natural phenomena and desirable, each of the other cases are not desirable and will have different solution strategies.

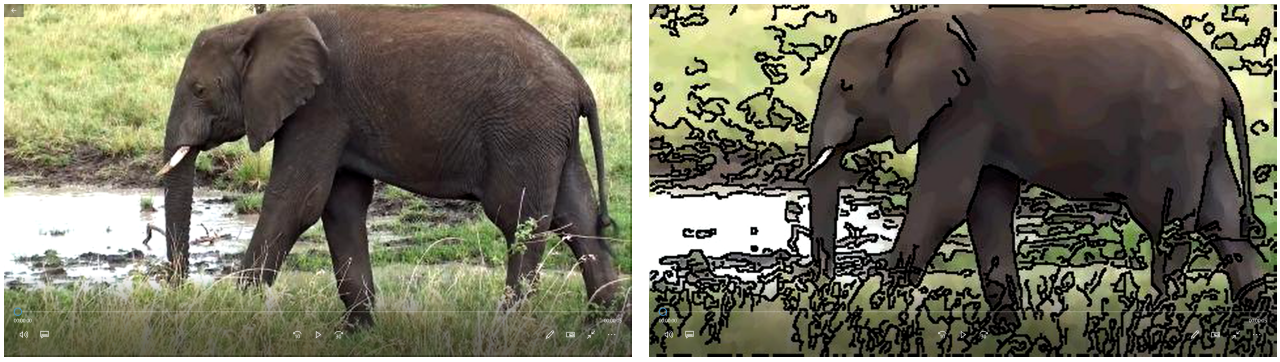

Figure 6: Natural Image vs Toonified Video

Figure 7: Two consecutive frames of a Toonified video

Conclusion

We were able to successfully replicate the results in the research paper. We see that this field is quite subjective and open to interpretations as to what actually is a "Toon". Still we attempted to find a common solution or maybe a solution which works in most cases.

Figure 8: Natural Image vs Toonified Image

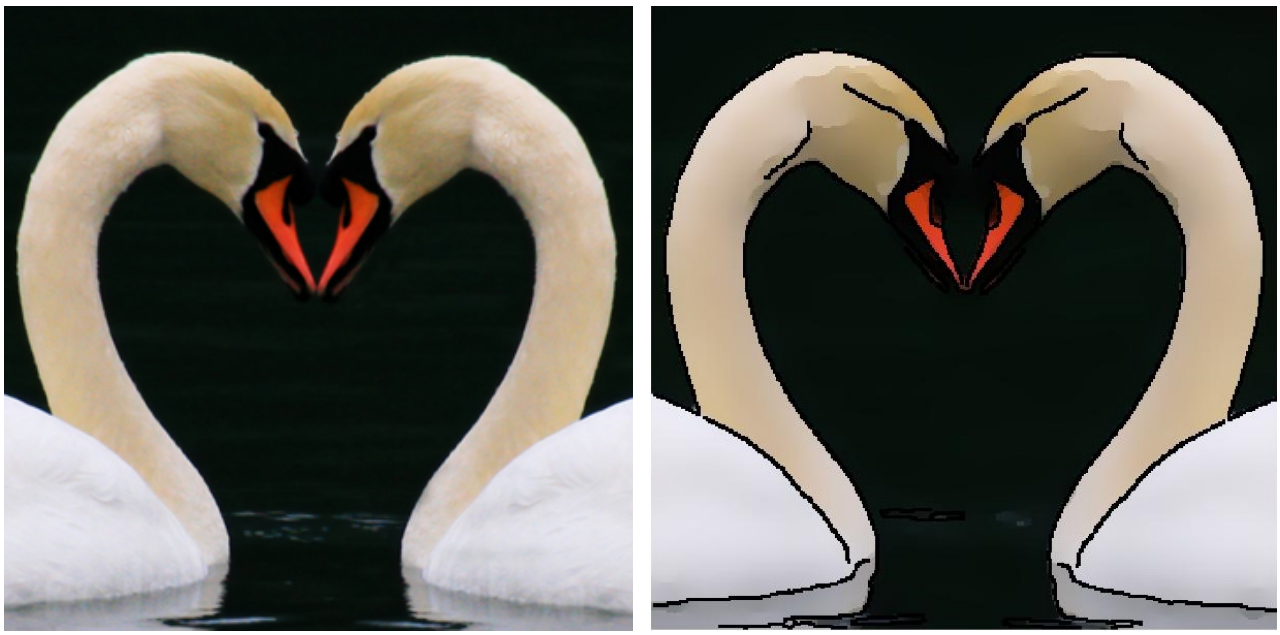

Figure 9: Natural Image vs Toonified Image

References

1. Bilateral Filtering
    - https://in.mathworks.com/help/images/ref/imbilatfilt.html
    - https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Tomasi98.pdf
    - https://en.wikipedia.org/wiki/CIELAB_color_space
2. https://en.wikipedia.org/wiki/Canny_edge_detector
3. Dataset is picked randomly from google images