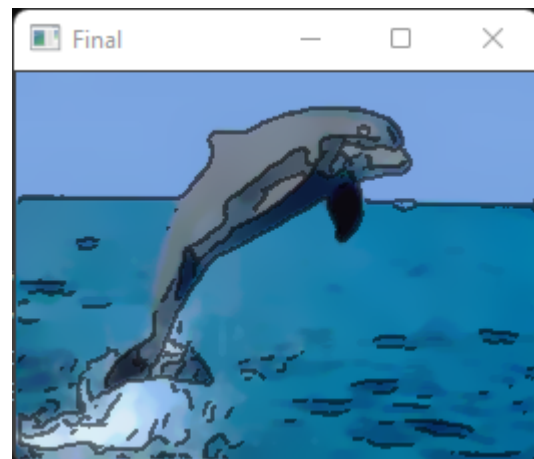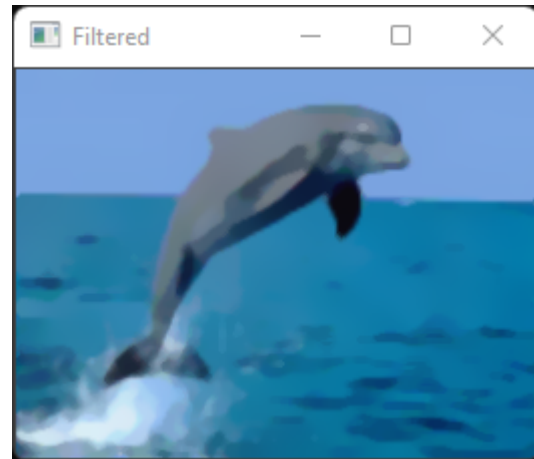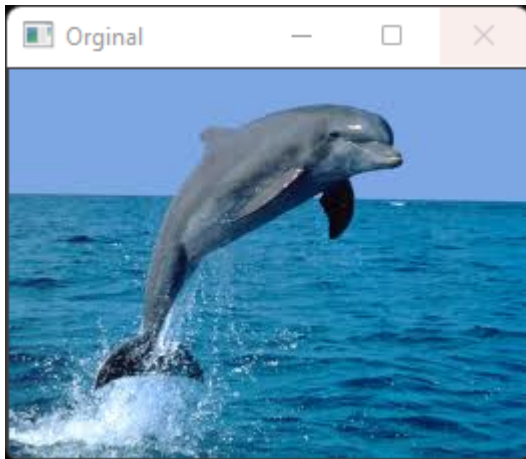# Toonify: Cartoon Photo Effect Application

Toonify seeks to emulate the types of cel-shading effects offered by graphics engines in a lighthearted and user friendly way.

Giurgiu Răzvan

30432

Image Processing

Project Goals

The goal of this project is to create an Open CV app with a simple user interface allowing users to apply the cartoon algorithm to images of their choice. The algorithm is designed to provide artistically and comically appealing results on as wide a range of pictures as possible, although it is conceded that not all inputs will yield equally satisfying results.

The Algorithm

The process to produce the cartoon effect is divided into two branches- one for detecting and boldening the **edges**, and one for **smoothing and quantizing** the colors in the image. At the end, the resulting images are **combined** to achieve the effect.

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. All edge processing tasks are performed with a single-channel grayscale image derived from the luminance values of the input.

Edge detection

Edge Detection: This algorithm uses the adaptive threshold edge detector. The benefits of using this algorithm is that we can check and select only edges that are over a certain threshold. This allows for morphological operators to be employed more predictably on the resulting edge image.

Morphological Operations:

Currently, we are using 2 morphological operations employed by the algorithm at this stage: dilation with a small structuring element and erosion with a 1 X 1 kernel The purpose of this step is to both bolden and smooth the contours of the edges slightly.

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 homogenous color regions. The colors in these regions are then requantized at a lower quantization.

Bilateral Filter

This filter is the key element in the color image processing chain, as it homogenizes color regions while preserving edges, even over multiple iterations. Because it is a computationally expensive task, the image is downsampled by a factor of 2 in both the x and y directions before being filtered. The bilateral filter works similarly to a Gaussian filter in that it assigns to each pixel a weighted sum of the pixel values in the neighborhood.

The bilateral filter runtime is dependent on the kernel size, and testing showed that running more iterations with a smaller kernel yielded results that were more aesthetically pleasing, and faster than those yielded by applying a bilateral filter with a large kernel size with less iterations. The Toonify algorithm uses a 7×7 kernel and iterates the filter 6 times on the image. Once the filtering is complete, the image is restored to it's original size.

Median Filter

A median filter as described above is applied after restoring the image to its full size in order to smooth over any artifacts that occurred during the upsampling. The kernel size is 3 × 3, as before, and the effect is hardly noticeable, except in some regions where there are tiny islands of a certain color located in a larger homogenous region of a different color.

Recombine

Once both the color and edge image processing chains are complete, the only task left is to overlay the edges onto the color image. It is possible to draw the edges in different colors on the image, and during development an algorithm was tested wherein each edge is drawn with a color based on its immediate surroundings, but this yielded unpredictable and aesthetically unappealing results. The final algorithm simply draws on all the contours in black.

Results

As with any algorithm that seeks to meet an aesthetic goal, measuring the success of Toonify poses some difficult problems. Specific goals throughout the process were to achieve solid continuous contours while avoiding small line clutter in the image, and also to achieve large regions of homogenous color, with a reduced color palette. Output images are evaluated on how well they meet these criteria. The following figures contain examples of desirable outputs and a result that does not meet the criteria.