Hacettepe University
Computer Engineering Department

BBM415 IMAGE PROCESSING LAB. - 2023 Fall

ASSIGNMENT 1

Giving Cartoon Effect to Colorful Images

Student name:

Hasan Malkoç

Instructor: Aydın Kaya
TA: Burçak Asal

Student Number:

2200356826

Giving Cartoon Effect to Colorful Images

Image filtering is a fundamental task in image processing, used for tasks like image smoothing and edge detection. Smoothing involves removing high-frequency components from an image using low-pass filters like the Gaussian filter. Edge detection aims to identify abrupt changes in brightness using filters like Sobel and Prewitt. Image editing tools, such as Photoshop, use special filters for artistic effects like cartoon or pencil drawing images. In studies like "Real-Time Video Abstraction" by Winnemoller et al., cartoon-like images are created by smoothing a color image with a non-linear filter, extracting edges, quantizing the smoothed image to reduce colors, and combining edges with the quantized image for the cartoon effect.

Approach

We'll be doing edge extraction with given steps in assignment description (Details, Edge Detection) after smoothing the image with either median or gaussian filtering with using allowed libraries. Then we are going to quantize our smoothed image with different values of quantization levels. After then, we are going to combine the extracted edges and quantized images and get a result of cartoon effect on colorful images.

Experiments:

1.Image smoothing

Median filtering

Original Image

Smoothed Image with kernel size of 3

Original Image

Smoothed Image with kernel size of 5

Original Image

Smoothed Image with kernel size of 8

Result: More the kernel size increases more the image gets blurred. Also my image gets black and white, I believe this is because of different techniques and applications of scipy.ndimage.median_filter.

Gaussian Filtering

Original Image

Smoothed Image with sigma value of 1.3

Original Image

Smoothed Image with sigma value of 2

Original Image

Smoothed Image with sigma value of 5

Result: More the sigma value more the image gets blurred. Since I got better results with gaussian filtering, I'll be using this instead of median filtering for the next experiments and result.

2. Edge Detection

Threshold Value Experiments:

```
sigma = 1.0
k = 1.6
threshold = 18
```

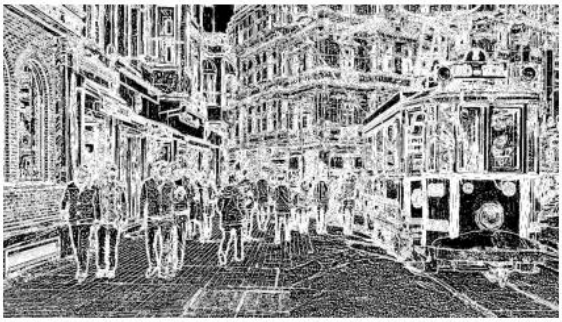
Original Image

Thresholded Image

```
sigma = 1.0
k = 1.6
threshold = 4
```

Original Image

Thresholded Image



Results: I kept the sigma and k values the same and change the threshold. When threshold gets smaller, my implementation gets more detailed edges and I don't want that for my cartoon effect. So, I might use relatively bigger value for thresholds. (like 15-20, It won't change too much between 15-200's)

Sigma Value Experiments:

```
sigma = 1.0
k = 1.6
threshold = 15
```

Original Image

Thresholded Image

```
sigma = 2.0
k = 1.6
threshold = 15
```

Original Image

Thresholded Image

```
sigma = 4.0
k = 1.6
threshold = 15
```

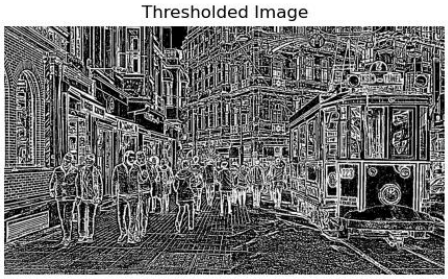
Original Image

Thresholded Image

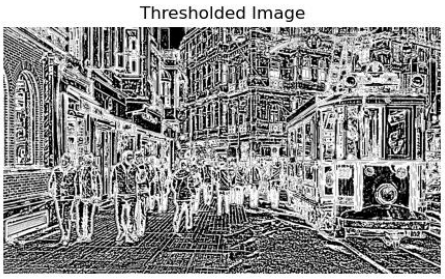
Results: If k and the threshold values remain the same, the larger sigma value may result in fewer detected edges. I can change this value for different pictures in order to get better results related to the edge details in the picture.

k Value Experiments:

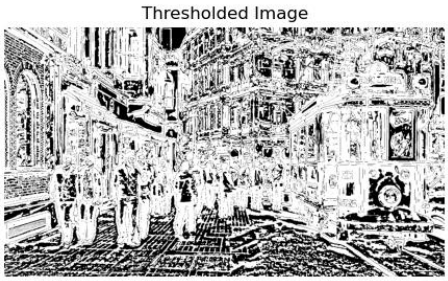
```
sigma = 1.3
k = 1.3
threshold = 15
```




```
sigma = 1.3
k = 2
threshold = 15
```




```
sigma = 1.3
k = 5
threshold = 15
```



Results: If sigma value and the threshold value remain the same, the larger sigma value may result noise in detected edges. When it gets larger it finds unnecessary edges like in the sky as seen in the picture. So for final results I'll be choosing smaller k values.

3. Quantization

Results: A higher quantization level means a larger step size, resulting in fewer distinct levels in the image and brings more coarsely quantized image. Higher level of quantization brings more simplified and low resolution image.

4. CARTOONIZE IMAGE

In this step I'll take the inverse of the estimated edges values and multiply it with the quantized image for each channel.

Original Image

Inverted Edges Image

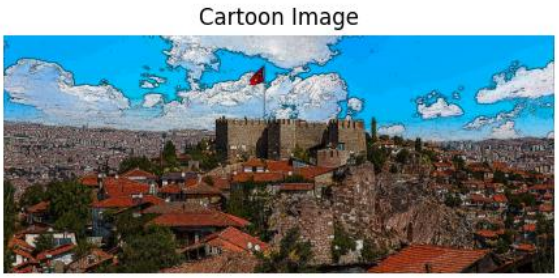
Original Image

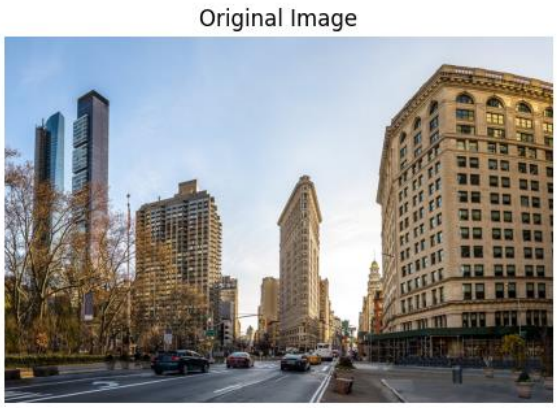
Cartoon Image

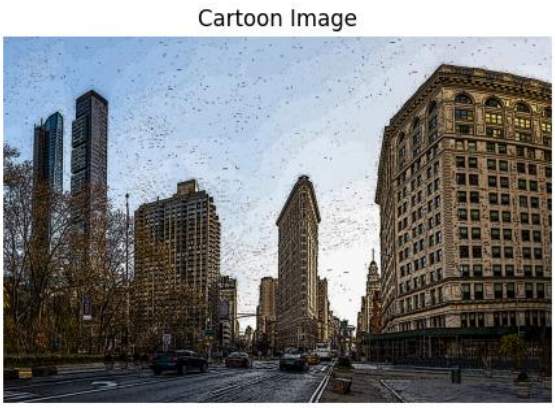
Result: even with an edge detailed image, my algorithm works very well. I might change some values like sigma, k, threshold, quantization level in the results to get better cartoonized images.

Some Final Results:

With random pictures:







Some pictures I took myself:

Original Image

Cartoon Image

Original Image

Cartoon Image

Original Image

Cartoon Image

Original Image

Cartoon Image

Original Image

Cartoon Image

Original Image

Cartoon Image

Also I tried some pictures of myself:

Original Image

Cartoon Image

Original Image

Cartoon Image

(more quality pictures give better results at it seems)

Original Image

Cartoon Image

(doesn’t work well on grass-like objects, it might be related with the noise)