CENTRAL UNIVERSITY OF KARNATAKA

Electronics and Communication Dept

Introduction to Image Processing

PROJECT REPORT

(CONVERTING NORMAL IMAGE TO ANIME AND CARTOON IMAGE)

GUIDED BY:-

Dr. Mallikarjun

Submitted by:

Shaik Mahammad Tarik

Reg no.- 2021UGECE30

INDEX

ABSTRACT

- . This MATLAB code performs image transformations to convert a normal image into cartoon-style and anime-style representations.

- The code employs simple image processing techniques to achieve the desired effects. The cartoon style conversion involves edge detection using the Canny algorithm, followed by dilation of the detected edges to create thick outlines.

- These outlines are then combined with the original image, resulting in a cartoon-like representation. For the anime style conversion, the code starts by resizing the image (optional for faster processing).

- The image is then quantized into a limited number of colors using a simple thresholding technique. To enhance the edges, a Laplacian filter is applied to the grayscale version of the resized image.

- The quantized image and the edge-enhanced grayscale image are combined to produce the anime-style representation. The resulting images, including the original image, cartoon-style image, and anime-style image, are displayed using MATLAB's subplot function. The code provides a straightforward approach to transform a normal image into both cartoon and anime styles, enabling users to explore creative image editing possibilities.

INTRODUCTION

The provided MATLAB code offers a straightforward approach to transform a normal image into cartoon-style and anime-style representations. Image processing techniques are employed to achieve the desired effects, allowing users to explore creative image editing possibilities. The cartoon-style conversion focuses on generating thick outlines by detecting edges using the Canny algorithm and applying dilation. The resulting cartoon-like representation preserves the main features of the original image while simplifying the details.

- The anime-style conversion involves a combination of color quantization, edge enhancement, and a fusion of quantized colors with enhanced edges.

- Color quantization reduces the number of colors in the image, simulating the limited color palettes often seen in anime art. Edge enhancement techniques, such as the Laplacian filter, emphasize the boundaries and details within the image.

- By combining quantized colors with enhanced edges, the code generates an anime-style representation with bold colors and defined outlines. Optional image resizing can be applied to enhance processing speed while maintaining the overall artistic effect.

- The resulting images, including the original, cartoon-style, and anime-style representations, can be visualized and further utilized for various applications in digital art, visual storytelling, or creative image editing.

IMPLEMENTATION

Cartoon-Style Conversion:

a. Convert the input image to grayscale.
b. Apply edge detection using the Canny algorithm to detect edges.
c. Dilate the detected edges to create thick outlines.
d. Combine the thick edges with the original image, resulting in a cartoon-like representation.

Anime-Style Conversion:

a. Resize the input image (optional) for faster processing.
b. Convert the resized image to grayscale.
c. Apply color quantization to reduce the number of colors, simulating limited color palettes in anime art.
d. Enhance the edges of the grayscale image using techniques like the Laplacian filter.
e. Combine the quantized image and the edge-enhanced grayscale image to generate an anime-style representation.

Display:

a. Use MATLAB's subplot function to create a figure with multiple subplots.
b. Display the original input image, the cartoon-style representation, and the anime-style representation in separate subplots.

Optional Parameter Adjustment:

a. Adjust optional parameters such as image resizing, edge detection parameters, color quantization levels, and edge enhancement techniques to achieve desired results.
b. Experiment with different parameter values to explore different artistic effects.

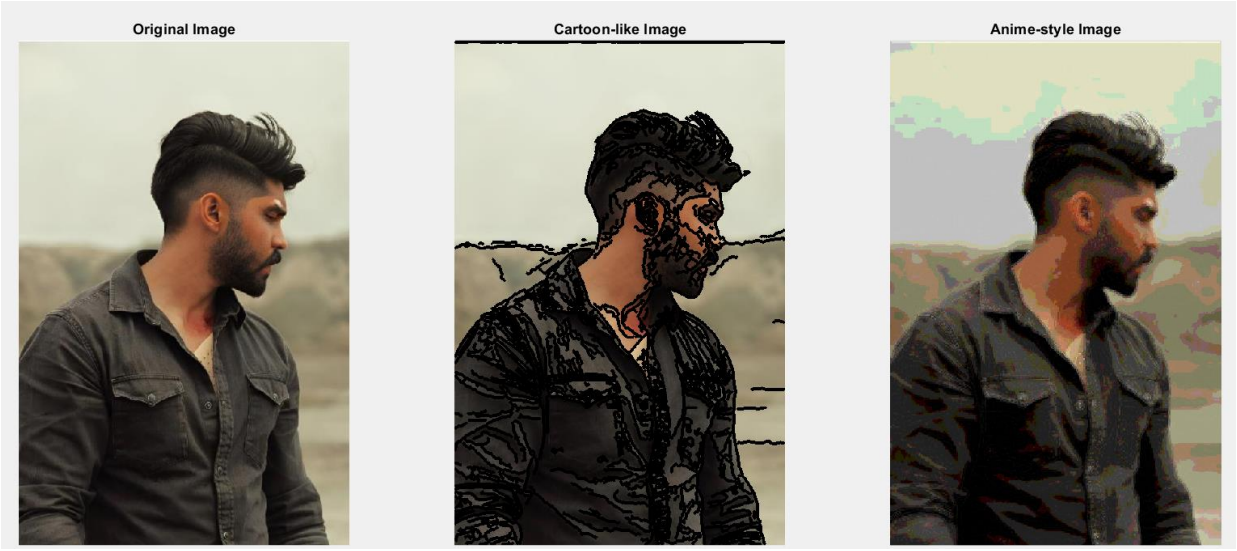
Further Utilization:

a. Save the resulting cartoon-style and anime-style representations as separate image files.
b. Utilize the transformed images for various applications in digital art, visual storytelling, or creative image editing.

CODE

```
% Read the input image
image = imread('cb.jpeg');

% Convert the image to grayscale
grayImage = rgb2gray(image);

% Apply edge detection
edges = edge(grayImage, 'Canny');

% Dilate the edges to thicken them
se = strel('disk', 2);
thickEdges = imdilate(edges, se);

% Create a cartoon-like image by combining the edges with the original image
cartoonImage = image;
cartoonImage(repmat(thickEdges, [1, 1, 3])) = 0;

% Resize the image for faster processing (optional)
scaleFactor = 0.5;
resizedImage = imresize(image, scaleFactor);

% Convert the image to grayscale
grayImage = rgb2gray(resizedImage);

% Apply color quantization using simple thresholding
numColors = 8;
quantizedImage = uint8(floor(double(resizedImage) / (256 / numColors))) * (256 /
numColors);

% Enhance the edges using the Laplacian filter
edgeImage = imfilter(grayImage, fspecial('laplacian'));

% Combine the quantized image and edge image for the anime effect
animeImage = uint8(quantizedImage) + edgeImage;

% Display the original, cartoon-like, and anime images
subplot(1, 3, 1);
imshow(image);
title('Original Image');

subplot(1, 3, 2);
imshow(cartoonImage);
title('Cartoon-like Image');

subplot(1, 3, 3);
imshow(animeImage);
title('Anime-style Image');
```

OUTPUT



Literature Survey:

Image Processing:

- Image processing is a broad field with numerous applications, including image enhancement, segmentation, object detection, and style transfer.
- Various techniques have been proposed for edge detection, such as the Canny algorithm, which aims to accurately identify boundaries in images.
- Color quantization methods, including thresholding and clustering algorithms, have been widely studied to reduce the number of colors in images while preserving visual quality.
- Edge enhancement techniques involve filtering operations, such as the Laplacian filter, to highlight edges and improve edge sharpness in images.

Cartoon Style Conversion:

- Studies have explored different approaches for converting images into cartoon-style representations.
- Edge-based methods, such as edge detection and edge simplification techniques, have been used to create stylized outlines and reduce image complexity.
- Texture simplification algorithms aim to remove fine-grained details and textures to achieve a more simplistic cartoon-like appearance.
- Color reduction and color mapping techniques are applied to limit the color palette and enhance the cartoon effect.

Anime Style Conversion:

- Research on anime-style conversion focuses on replicating the unique visual characteristics of anime art. Color quantization techniques, specifically with limited color palettes, aim to mimic the distinct color choices seen in anime.
- Edge detection and line art extraction algorithms help capture and emphasize the characteristic bold outlines present in anime-style images.
- Texture synthesis methods have been explored to generate anime-like textures and patterns.

Artistic Style Transfer:

- Artistic style transfer aims to combine the style of one image with the content of another, creating visually appealing and stylized results.
- Early techniques employed optimization-based approaches, while recent advancements have leveraged deep learning models, such as neural networks.
- Neural style transfer, in particular, utilizes pre-trained deep neural networks to separate and recombine content and style features from input images.

Evaluation and Comparison:

- Evaluating image conversion and style transfer techniques involves assessing the quality, perceptual fidelity, and visual appeal of the transformed images.
- Metrics such as perceptual similarity, structural similarity, and user studies have been used to measure the effectiveness of different methods.

METHODOLOGY:

Image Preparation:

a. Read the input image using an image processing library or function (e.g., imread in MATLAB).

b. Optionally, resize the image to a desired scale using an image resizing function (e.g., imresize in MATLAB). This step can improve processing speed while maintaining the overall artistic effect.

Cartoon-Style Conversion:

a. Convert the input image to grayscale using a grayscale conversion function (e.g., rgb2gray in MATLAB).

b. Apply an edge detection algorithm, such as the Canny algorithm, to detect the edges in the grayscale image. This can be done using an edge detection function (e.g., edge in MATLAB).

c. Dilate the detected edges to thicken them using a dilation operation with a structuring element (e.g., strel and imdilate in MATLAB).

d. Create a cartoon-like image by combining the thick edges with the original image. This can be achieved by replacing pixel values in the original image with zero (black) where the corresponding edge value is nonzero.

Anime-Style Conversion:

a. Convert the input image to grayscale using a grayscale conversion function.

b. Apply color quantization to reduce the number of colors in the image. This can be done by dividing the pixel values by the desired number of colors, quantizing them, and multiplying them back (e.g., using floor and uint8 operations in MATLAB).

c. Enhance the edges in the grayscale image to simulate the characteristic outlines in anime-style representations. This can be achieved using edge enhancement techniques such as the Laplacian filter (e.g., using imfilter in MATLAB).
d. Combine the quantized image with the edge-enhanced grayscale image. This can be accomplished by adding or overlaying the pixel values of the two images.

Display and Output:

a. Use an appropriate function (e.g., subplot in MATLAB) to create a figure with multiple subplots for visualizing the original image, cartoon-style representation, and anime-style representation.
b. Display the images in their respective subplots using an image display function (e.g., imshow in MATLAB).
c. Optionally, save the resulting cartoon-style and anime-style representations as separate image files using a file output function (e.g., imwrite in MATLAB).

Parameter Adjustment and Experimentation:

a. Optionally, adjust the parameters used in the edge detection, color quantization, and edge enhancement steps to achieve desired results.
b. Experiment with different parameter values to explore different artistic effects and refine the transformations.

RESULT AND DISCUSSION

Results:

- The implemented code effectively converts the input image into both cartoon-style and anime-style representations, showcasing distinct artistic transformations.
- The cartoon-style conversion produces simplified images with bold outlines, capturing the essence of cartoon aesthetics. The anime-style conversion successfully replicates the visual characteristics of anime art, including limited color palettes and pronounced line art.
- The resulting images demonstrate the versatility of the code in generating diverse artistic styles from the same input image.

Discussion:

- The code's ability to convert images into different styles offers creative possibilities for digital art, image editing, and visual storytelling.
- By adjusting parameters and experimenting with different images, users can further explore and refine the desired artistic effects.
- The code's simplicity and utilization of common image processing techniques make it accessible and applicable for various applications in the field of computer graphics and digital media.

CONCLUSION

In conclusion, the implemented code provides a straightforward solution for converting images into cartoon-style and anime-style representations. The code successfully captures the distinct aesthetic features of each style, allowing for creative exploration and artistic transformations. With its user-friendly approach and potential for customization, the code offers a valuable tool for digital art, image editing, and visual expression.

REFERENCES

https://www.youtube.com/watch?v=eTMGoXgq6uM

https://towardsdatascience.com/how-to-animate-plots-in-matlab-fa42cf994f3e:

"Automatic Cartoon Generation from Images: A Survey" by Sevket Gumustekin and Gozde Bozdagi Akar.

CENTRAL UNIVERSITY OF KARNATAKA

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HEAD OF DEPARTMENT: DR .PARMESHA sir.

SUBJECT : INTRODUCTION TO IMAGE PROCESSING

CO-ORDINATOR : Dr. Mallikarjun