

CARTOONATOR



TEAM MEMBERS

SAHIL BANSARI

GAURAV VISHWAKARMA

PARTH SHARMA

SANJANA KUMARI

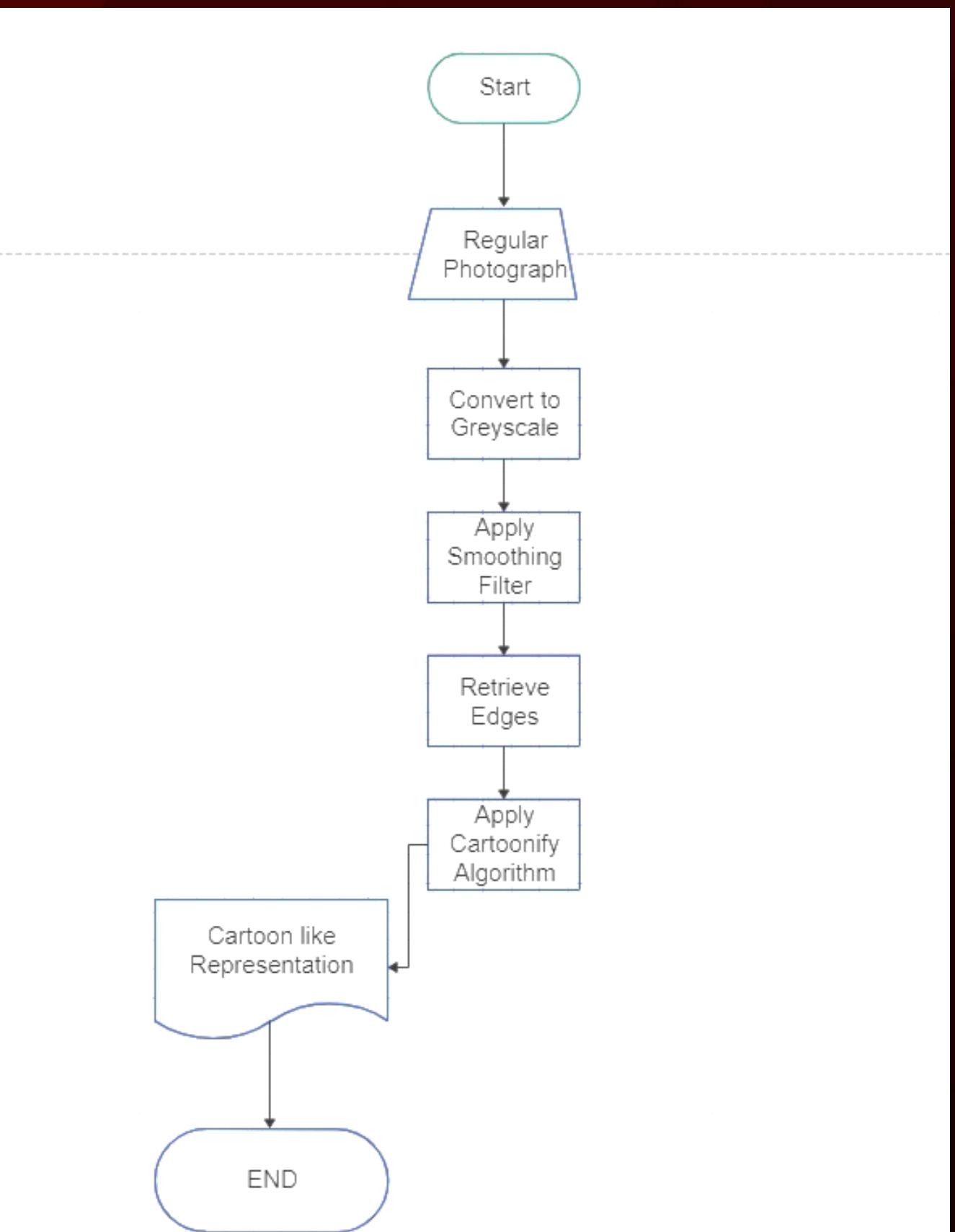
TANUSHKA PAWAR

ATHARV WAGH

INTRODUCTION

In today's digital landscape, the demand for streamlined image editing tools is undeniable. Cartoonization, a popular technique for transforming photos into captivating illustrations, often falls short due to limited quality and flexibility in existing methods. To address this, we introduce an innovative platform focused on revolutionizing cartoonization through Python. By blending traditional picture processing with smart deep learning, our platform offers an intuitive interface and superior results. It represents a significant leap forward in image editing, empowering users to express their creativity effortlessly.

PROJECT ARCHITECTURE



IMPLEMENTATION DETAILS

Preprocessing and Model Initialization:
Preprocess input images by resizing and cropping to match model requirements.

Inference and Output Generation:

- Feed preprocessed input image into the model for cartoonization.
- Apply post-processing to adjust pixel values for visual quality.

Demonstration and Clean-up:
Showcase tool's functionality with a sample image cartoonization.

SAMPLE EDIT

INPUT IMAGE:



CARTOONIZED IMAGE



FUTURE SCOPES

- **Watercolor Painting Style:**
- **Develop algorithms to simulate the soft blending and fluidity characteristic of watercolor paintings, offering users the ability to transform images into dreamy and ethereal artworks reminiscent of traditional watercolor techniques.**

- **Pointillism Style:**
- **Implement techniques to create compositions using distinct dots or small marks, mimicking the intricate patterns of pointillist paintings. Users can transform images into visually striking artworks composed of points of varying colors and sizes.**



FUTURE SCOPES

- **Binary Style:**
- **Explore algorithms to convert images into binary representations, where each pixel is either black or white based on a specified threshold. This style offers a unique and minimalist aesthetic, allowing for creative expression within the constraints of binary representation.**
- **Cubist Style:**
- **Develop methods to deconstruct and reconstruct images into geometric shapes and abstract forms, resembling the unconventional artworks of cubist painters. Users can transform images into visually dynamic compositions with distorted perspectives and fragmented elements.**



CONCLUSION

- In conclusion, our program offers an innovative solution for transforming photographs into captivating artworks. Through the integration of Python and OpenCV, we provide users with a versatile platform to explore various artistic styles, from watercolor painting to cubist representations. With a user-friendly interface and robust performance, our program empowers users to unleash their creativity effortlessly. We are committed to ongoing improvements, ensuring continued innovation in image editing and artistic expression.

REFRENCES



1. S. Rajatha, A. S. Makkigadde, N. L. Kanchan, Sapna, and K. J. Bhat, "Cartoonizer: Convert Images and Videos to Cartoon-Style Images and Videos", IJRESM, vol. 4, no. 7, pp. 275–278, Jul. 2021, Accessed: May 03, 2024.
2. Saurabh Kumar, Utkarsh Bhardwaj and T. Poongodi, "Cartoonify an image using open cv in python", 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), pp. 952-955, 2022.
3. Leon A. Gatys, Alexander S. Ecker and Matthias Bethge, "Image style transfer using convolutional neural networks", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414-2423, 2016.
4. Fatin Alkinani and Abdul Monem Rahma, "Enhanced cartooning system based on dynamic augmented kmcg.", Iraqi Journal of Science, vol. 60, pp. 653-661, 2019.
5. Y. Chen, Y.-K. Lai, Y.-J. Liu, "CartoonGAN: Generative Adversarial Network for photo cartoonization", International Conference on Image Processing, 2018.



THANK YOU FOR WATCHING