Major Project Report

Submitted in partial fulfillment of the degree of

B.Tech

By

Shweta Bhagat (11900120090)

3rd year, 5th-semester student of

Siliguri Institute of Technology

Under the supervision of

Mr. Mainak Deb

SikharthyInfotechPvt. Ltd.

Department of Computer Science

Date: 16.09.22

I as a result of this forward the documentation prepared under my supervision by **Shweta Bhagat** entitled **"Cartoonify Image Using OpenCV"** is accepted as fulfillment of the requirement for the Degree of B.Tech in **Computer Science and Engineering** from **Siliguri Institute of Technology** affiliated with **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**) .

___________________________ ________________________________

Mr. Mainak Deb
Project Guide
SikhartyinfotechPvt.Ltd.

Shweta Bhagat
Computer Science and Engineering
Siliguri Institute of Technology

Cartoonify Image Using OpenCV in Python

By

Shweta Bhagat (11900120090)

Under the guidance of

Mr. Mainak Deb

Project Guide

SikharthyInfotechPvt. Ltd.

THEIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

B.Tech

IN

COMPUTER SCIENCE AND ENGINEERING

SILIGURI INSTITUTE OF TECHNOLOGY

AFFILIATED TO

Maulana Abul Kalam Azad University of Technology

Address: Hill Cart Road, Salbari, Sukna,Siliguri, West Bengal

West Bengal-743009

Phone: 09434527272

Email:*register@sittechno.org*

Certificate of Approval

The foregoing project is hereby approved as a creditable study for the B.Tech in Computer Science and Engineering presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approved any statement made, opinion expressed, or conclusion therein but approves this project only for the purpose for which it is submitted.

Final Examination for
Evaluation of the Project ---------------------------------------

----

Signatures of Examiners

ABSTRACT

The purpose of the project entitled **“Cartoonify Image Using OpenCV”** is to convert a user input image to cartoonified image with the help of OpenCV, Tkinter, Python, and its libraries. It also allows the user to save the image to their system.

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals in this attempt. Our project report has been structured under the valued suggestion, support, and guidance of **Mr.Mainak Deb**. Under his guidance, we accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

Shweta Bhagat

Computer Science and Engineering

TABLE OF CONTENTS

Chapter 1: Introduction

1: Introduction

Chapter 2: What We Used

2.1: Python

2.2: Libraries

2.3: IDE

Chapter 3:Functionality and Features

3.1: Upload Image

3.2: Save Image

Chapter 4: Algorithm and Steps to Cartoonify Image

Chapter 5:Project Planning and Scheduling

Chapter 6: Functional Requirement of the System

6.1: Hardware Requirement

6.2: Software Requirement

Chapter 7: Conclusion

Chapter 8: References

INTRODUCTION

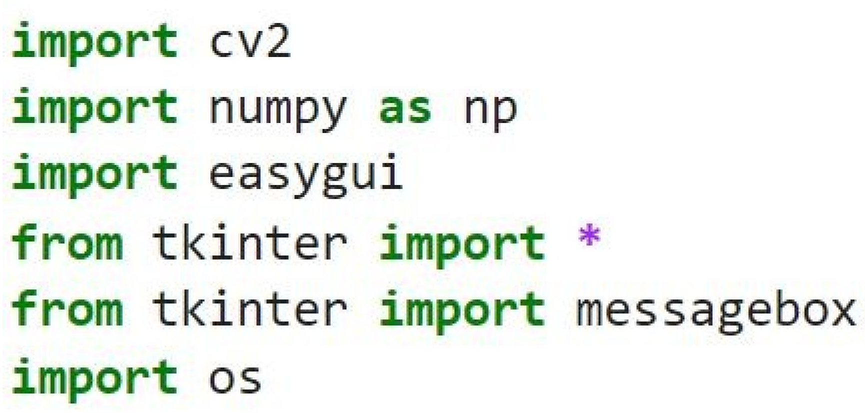
With the recent success of Instagram, the popularity of simple and fun photo effects apps have been on the rise. The mobile platform presents a unique arena for these applications by connecting users with both the means to capture images and the computational power to perform sophisticated processing on these images. Cartoonify seeks to leverage the existing OpenCV library in order to emulate cartoon effects.

2. WHAT WE USED

2.1 PYTHON

We used python 3.10.5, in this project.Its because it has vast libraries that we have used in our project.

2.2 LIBRARIES



```
import cv2
import numpy as np
import easygui
from tkinter import *
from tkinter import messagebox
import os
```

1.OpenCV

It is a cross-platform library used for Computer Vision. It includes applications like video and image capturing and processing. It is majorly used in image transformation, object detection, face recognition, and many other stunning applications.

2. Tkinter

It is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

3. NumPy

It is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. NumPy stands for Numerical Python.

4. EasyGUI

It is a module for very simple, very easy GUI programming in Python. EasyGUI is different from other GUI libraries in that EasyGUI is NOT event-driven. Instead, all GUI interactions are invoked by simple function calls.

Python OS

This module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about the operating system.

2.3 IDE:- Visual studio code

3. FUNCTIONALITY and FEATURES

3.1 UPLOAD IMAGE

Users will be able to upload their desired image from their system, which will be then fed to the program to give it a cartoon effect.

3.2 SAVE IMAGE

After an image gets converted to its cartoonfied form, users will be able to save it to their system.

Chapter 4: Algorithm and Steps to Cartoonify Image

Edges : Finding smooth continuous contours is an important component of 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.

Median Filter: before any further processing, a median filter is applied in order to reduce any salt and pepper noise that may be in the image. At this stage, the median filter kernel is a 7×7 matrix with the centroid as the central element in the matrix. The median filter is small enough to preserve edges in the image, which is important. Too large a kernel size would result in washed-out edges.

Edge Detection: This algorithm uses the popular Canny edge detector. The benifits of using the Canny edge detector instead of a Laplacian kernel is that the edges are all single pixel edges in the resulting image. This allows for morphological operators to be employed more predictably on the resulting edge image.

Morphological Operations: Currently, the only morpho-logical operation employed by the algorithm at this stage is dilation with a small 2 × 2 structuring element. 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 quantized 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 4 in both the x and y directions before being filtered. This algorithm uses a 9×9 kernel and iterates the filter 14 times on the image. Once the filtering is complete, the image is restored to its original size using linear interpolation to fill in the missing pixels.

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 up-sampling. The kernel size is 7 × 7, 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. The final algorithm simply draws on all the contours in black of the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are:

6. Project Planning and Scheduling

Scheduling is an important activity of any project management. Scheduling a software project involves first breaking down an entire problem into a logical set of tasks that would be assigned to developers.

In order to Schedule our project we have to do the following:

▪ Identify the tasks needed to complete the project.

  - Creating the main window.
  - Adding functionality to the window to open and save images.
  - Implementing the cartoonify algorithm in OpenCV

▪ Determine the dependency among different tasks.

  - For creating a window we need Tkinter.
  - For adding file opening and saving functionality we use easygui.
  - To implement the caartonize algorithm we use OpenCV.

7. Functional Requirement of the System

7.1 HARDWARE REQUIREMENTS

The minimum Hardware requirements for the application to run smoothly should have the following configuration:

Processor	Intel Core i3
RAM	4GB or more
HDD	3GB or more

7.2 SOFTWARE SPECIFICATIONS

The minimum software requirements are as follows:

Operating System	Windows 8 and upwards
Language Used	Python
Working IDE	Visual Studio Code

9. CONCLUSION

We have made an application that allows users to cartoonify an image of their choice and save it on their system. Our application makes use of the OpenCV library for processing the image. With the help of Tkinter, we were able to make a GUI for our application. During the development process, we came to know about image processing and how it works.

10. REFERENCES

1. OpenCV Python Tutorial - GeeksforGeeks

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