

IMAGE CARTOONIZATION

A Course Based Project Submitted in Partial Fulfilment of the Requirement for the
Award of the degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

Submitted by

22075A6206-VAIDYA ABHIJEETH KUMAR



DEPARTMENT OF CSE-CYS, DS & (AI &DS)

VALLURUPALLI NAGESWARARAO VIGNANA JYOTHI

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute, NAAC Accredited With 'A++' Grade, NBA
Accredited, Approved by AICTE, New Delhi, Affiliated to JNTUH)

VALLURUPALLI NAGESWARARAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute)



CERTIFICATE

This is to Certify that VAIDYA ABHIJEETH KUMAR (22075A6206) has successfully completed their project work at CSE CYS, DS & (AI & DS) Department of VNRVJIET, Hyderabad entitled “IMAGE CARTOONIZATION” in partial fulfilment of the requirements for the award of the Bachelor of Technology degree during the Academic year 2022-2023

Project Guide

Mrs. E. Lalitha
Assistant Prof. and Internal Guide
Dept. of CSE-CYS, DS and AI&DS
VNRVJIET

Head of Department

Dr. M. Rajasekhar
Prof. and Head
Dept. of CSE-CYS, DS and AI&DS
VNRVJIET

DECLARATION

This is to certify that the project work entitled "IMAGE CARTOONIZATION" submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfilment of requirement for the award of Bachelor of Technology in Computer Science and Engineering. It is a Bonafide report of the work carried out by us under the guidance and supervision of Mrs.E.Lalitha (Assistant Professor), Department of CSE-CYS,DS,AI&DS, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

VAIDYA ABHIJEETH KUMAR
22075A6206
CSE-CYBERSECURITY

ACKNOWLEDGEMENT

Behind every achievement lies the heartfelt gratitude to those who activated in completing the project. To them we lay the words of gratitude within us.

We are indebted to our venerable principal Dr. C.D. NAIDU for this inflicting devotion, which led us to complete this project. The support, encouragement given by him and his motivation led us to complete the project.

We express our sincere thanks to internal guide Mrs.E.Lalitha and also Head of the Department Dr. M. RAJA SHEKHAR for having provided us a lot of facilities to undertake the project work and guide us to complete the project.

We take the opportunity to express thanks to our faculty of the Dept. of COMPUTER SCIENCE AND ENGINEERING-CYBER SECURITY and remaining members of our college VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY who extended their valuable support in helping us to complete the project in time.

VAIDYA ABHIJEETH KUMAR 22075A6206

ABSTRACT

In this project, we will build one interesting application that will cartoon the image provided to it. To build this cartoon application we will use python and OpenCV. This is one of the exciting and thrilling applications of Machine Learning. While building this application we will also see how to use libraries like easygui, Tkinter, and all. Here you have to select the image and then the application will convert that image into its cartoon form. Mainly, we build this application using OpenCV and python as the programming languages.

CONTENTS

Topics	Page No.
1. ACKNOWLEDGEMENT	4
2. ABSTRACT	5
3. INTRODUCTION	7
4. LIBRARIES	8
5. CODE	9-10
6. OUTPUT	11
7. CONCLUSION	12

INTRODUCTION

OpenCV is an open-source library in python that is used mainly for computer vision tasks in the areas of machine learning and artificial intelligence. Nowadays, openCV is playing a major role in the field of technology. Using OpenCV we can process images and videos for some tasks like object detection, face detection, object tracking, and all.

Requirements

Python: We use python as a programming language for building the application.

cv2: We use cv2 for image processing.

LIBRARIES

Numpy: Mainly NumPy is used for dealing with arrays. Here the images that we use are stored in the form of arrays. So for that, we use NumPy.

easygui: easygui is a module used for GUI programming in python. In our application easygui is used to open the file box to upload images from the local system.

We create a function cartoonify which includes all the steps from converting to greyscale to the final cartoon image. The first step is to convert it into a greyscale image and then apply the blur to smoothen the image which is one of the main steps in cartooning the image. For smoothening, the blur effect is given using the median blur() function.

and then retrieving the edges and highlighting for cartoon effect which is another important step of the application and then applying the bilateral filter which is in built that helps to remove the noise

present in the image and provides the clean image and then masking the edged image and Finally we plot an image which contains all the six traditions throughout the process.

CODE

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from google.colab import files
def read_file(filename):
    img = cv2.imread(filename)
    cv2_imshow(img)
    return img
def color_quantization(img, k):
# Transform the image
    data = np.float32(img).reshape((-1, 3))

# Determine criteria
    criteria = (cv2.TERM_CRITERIA_EPS +
cv2.TERM_CRITERIA_MAX_ITER, 20, 0.001)

# Implementing K-Means
    ret, label, center = cv2.kmeans(data, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
    center = np.uint8(center)
    result = center[label.flatten()]
    result = result.reshape(img.shape)
    return result
def edge_mask(img, line_size, blur_value):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray_blur = cv2.medianBlur(gray, blur_value)
    edges = cv2.adaptiveThreshold(gray_blur, 255,
cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, line_size,
blur_value)
    return edges

uploaded = files.upload()
```

```
filename = next(iter(uploaded))
img = read_file(filename)
line_size = 7
blur_value = 7

edges = edge_mask(img, line_size, blur_value)
cv2_imshow(edges)
total_color = 9

img = color_quantization(img, total_color)
cv2_imshow(img)
blurred = cv2.bilateralFilter(img, d=7, sigmaColor=200,sigmaSpace=200)
cv2_imshow(blurred)
cartoon = cv2.bitwise_and(blurred, blurred, mask=edges)
cv2_imshow(cartoon)
```

Input:



Output:



Conclusion:

Finally, we will get the output image as shown above. It contains all the 4 transitions of the image. And the final image is the cartoon image. I hope you have enjoyed this application. This is the “Cartoon of an image Version”. Now using this application, you can create your cartoon image.