**PROJECT REPORT**

ON

**“Cartoonify Image With QR Scanner”**

*(using Python and Machine Learning)*



##### 

##### *Submitted by*

MANALI SAHU

***in partial fulfillment for the award of the degree of***

#### BACHELOR OF ENGINEERING

**IN**

**COMPUTER SCIENCE & ENGINEERING**

Under the Supervision of- Submitted by:

**Mrs.Ruchi Talwar** Manali Sahu (Assistant Professor) (201500380)

Acknowledgment

It gives us a great sense of pleasure to present the synopsis of the B.Tech mini project (**Cartoonify Image**) undertaken during B. Tech III Year. This project is going to be an acknowledgment of the inspiration,drive, and technical assistance that will be contributed to it by many individuals.

I owe a special debt of gratitude to **Mrs.RuchiTalwar(Assistant Professor Department of CEA),** for providing me with an encouraging platform to develop this project, which thus helped in shaping my abilities towards a constructive goal, through her constant support and guidance to our work.

Her thoroughness and perseverance have been a constant source of inspiration for me. I believe that she will support me with all her experienced ideas and insightful comments at different stages of the project & also teach me about the latest industry-oriented technologies.

I also do not like to miss the opportunity to acknowledge the contribution of all department faculty members for their kind guidance and cooperation.

By

Manali Sahu

(201500380)

|  |  |
| --- | --- |
| ABSTRACT  The project aims to put forward a solution for transforming images into animated photos(Cartoon Images). The earlier method of transformation requires complicated computer graphics and skills. The idea of the project is based on designated images that are converted into an art form such as painting. Cartoonify seeks to leverage the existence of the Opencv2 library.  Image process tools include OpenCV, Scikit Image, and Numpy.  The algorithm is designed to provide artistically and comically appealing results on a wide range of pictures as possible, although it is conceded that not all inputs yield equally satisfying results. Edge detection and the concept of filters have been used. Matplotlib (library) is used for visualization and plotting. Thus it is imported to form the plot images. This work can be extended to real-time video in the future. An additional feature is added is that we can scan the output image through QR Code. | |
|  |

##### Computer Engineering and Application

**GLA University, Mathura**

#### BONAFIDE CERTIFICATE

Certified that this project report **“Cartoonify Image With QR Scanner”** is the bonafide work of **“Manali Sahu”** who carried out the project work under my/our

|  |  |
| --- | --- |
| supervision.  Mentor: **Ruchi Talwar**  Signature:  Submitted for the project :**24-November-2022**  **INTERNAL EXAMINER :**    **EXTERNAL EXAMINER:** |  |

Declaration

We hereby declare that the work presented in this dissertation entitled “Cartoonify image with QR Scanner” has been done by me under the guidance of our mentor **Mrs.Ruchi Talwar**, and this dissertation embodies my work.

BY :

MANALI SAHU

201500380

TABLE OF CONTENT

Page no.

1. INTRODUCTION OF THE PROJECT…………………………………8
2. ABOUT THE PROJECT………………………………………………..9-11
   * 1. Python Language
     2. OpenCV
     3. Libraries used in the project
     4. Platform used ---- Google colab
     5. QR Scanner
3. OBJECTIVE OF THE PROJECT……………………… .………………12
4. WHY WE NEED THE PROJECT……………………………………….12
5. ALGORITHM USED …………………………………………………….13
6. DIAGRAM & EXPERIMENTALANALYSIS………………………….14-18
   * + 1. User Diagram
       2. Flow chart
       3. Some examples of the project
       4. Graph used
7. CODE DESCRIPTION…………………………………………………..19-20
8. MOTIVATION……………………………………………………………21
9. FUTURE PROSPECT……………………………………………………..21
10. CONCLUSION …………………………………………………………..22
11. BIBLOGRAPHY……………………………………………………..22

INTRODUCTION

In this, we will build one interesting application that will cartoonify the image provided to it. To build this cartoonifyer application we will use python and OpenCV. This is one of the exciting and thrilling applications of Machine Learning. While building this application we have used various libraries. 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.

The algorithm is designed to provide artistically and comically appealing results on a wide range of pictures as possible, although it is conceded that not all inputs yield equally satisfying results. Edge detection and the concept of filters have been used.

In our project we have also used QR Scanner for scanning the result by using url of that image .it is very useful and time efficient







**Sketch Image**

**Original Image**

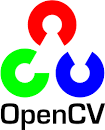
**Cartoon Image**

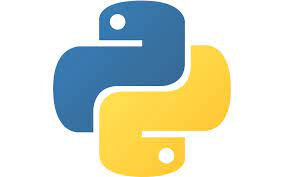
ABOUT THE PROJECT

**OpenCV:** Python is the pool of libraries. It has numerous libraries for real-world applications. One such library is OpenCV. OpenCV 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.

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.

OpenCV has c, c++, java, and python interfaces and it supports all kinds of systems such as Windows, Linux, Android, Mac OS, IoS, and all.



**PYTHON:** **Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. . It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python supports multiple programming paradigms, including Procedural, Object Oriented, and Functional programming languages.

**NUMPY LIBRARIES**

NumPy in Python is a library that is used to work with arrays and was created in 2005 by Travis Oliphant. NumPy library in Python has functions for working in the domain of [Fourier transform](https://www.mygreatlearning.com/academy/learn-for-free/courses/fourier-transform-explained?gl_blog_id=23244), linear algebra, and matrices. Python NumPy is an open-source project that can be used freely. NumPy stands for Numerical Python.

Installing the NumPy library is a straightforward process. You can use pip to install the library.

**Gaussian Filter**

A Gaussian Filter is a low pass filter used for reducing noise (high-frequency components) and blurring regions of an image. The filter is implemented as an Odd sized Symmetric Kernel (DIP version of a Matrix) which is passed through each pixel of the Region of Interest to get the desired effect.

**Median Filter**

Median filter is one of the well-known order-statistic filters due to its good performance for some specific noise types such as “Gaussian,” “random,” and “salt and pepper” noises [3]. According to the median filter, the center pixel of a M × M neighborhood is replaced by the median value of the corresponding window. Note that noise pixels are considered to be very different from the median.

**PIL Library**

Python Imaging Library (expansion of PIL) is the de facto **image processing package for Python language**. It incorporates lightweight image processing tools that is used in editing, creating, and saving images. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

**Google Colab Platform :**

Collaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser and is especially well suited to machine learning, data analysis, and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources.

* Write and execute code in Python
* Document your code that supports mathematical equations
* Create/Upload/Share notebooks
* Import/Save notebooks from/to Google Drive
* Import/Publish notebooks from GitHub

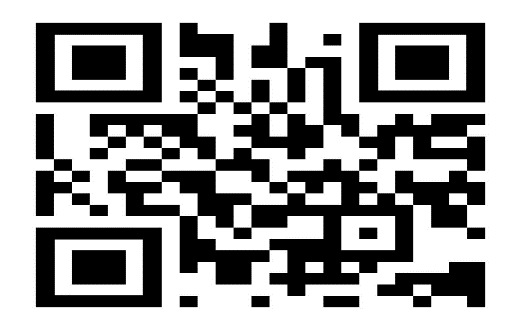


**QR Scanner:**

A QR code is a type of barcode that can be read easily by a digital device and which stores information as a series of pixels in a square-shaped grid. QR codes are frequently used to track information about products in a supply chain and – because many smartphones have built-in QR readers – they are often used in marketing and advertising campaigns.

With an online QR code generator, you can create QR codes readable in both vertical and horizontal dimensions. QR codes can store various data and information, including a link to an online store, an influencer's social media sites, and files such as photos, videos, music, official documents, and more.

QR codes are popular because you don't need the technical know-how to scan them with your mobile device. It's important to know that Android and iOS perform QR code scanning differently.



OBJECTIVE OF THE PROJECT

The main objective is to put ahead a result for making over pictures or vids of real- global into animated prints (Cartoon Images) or Videotape.

This project would follow an OpenCV library within python whilst making the layout. Python is a collection of libraries and it has multiplex libraries for real-world operations. One corresponding library is OpenCV and it is a move-platform library used for Computer Vision. It consists of operations like vid and photo taking pictures and processing.

WHY I HAVE CHOOSEN THIS PROJECT ?

The idea behind choosing this project was to create a cartoon from a picture. This project includes areas of vision (object recognition) and carnification. This was thought to be a step in helping the children to better understand and appreciate cartoons. Cartoons seem to be very familiar to children so they can relate to them.

Social media is extensively used these days. Standing out in this online crowd has always been a to-do for every user’s list on these social media platforms. Be it images, blog posts, artwork, tweets, memes, opinions, and what not being used to seek the attention of followers or friends to create influence or to connect with them on such social platforms. We aim to provide one such creative solution to their needs: applying cartoon-like effects to their images

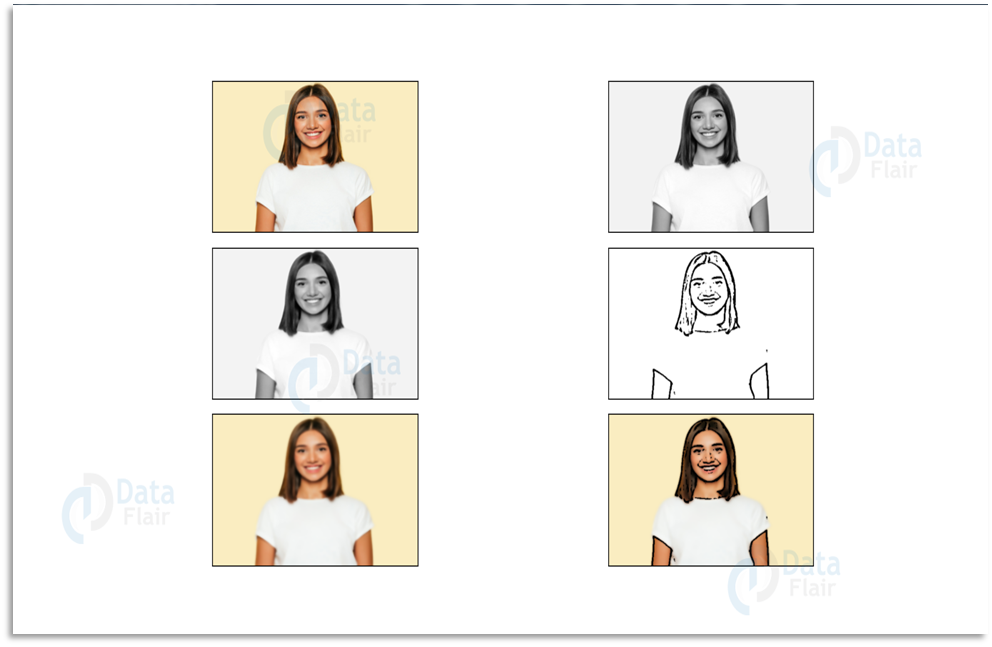
ALGORITHM USED

* Step 1: Importing the required modules:
* OpenCV, Numphy, OS, MATPOLIB
* Step 2: Building a File Box to choose a particular File
* Step 3: Transforming an image to grayscale
* Step 4: Smoothening a grayscale image: using MedianBlur()

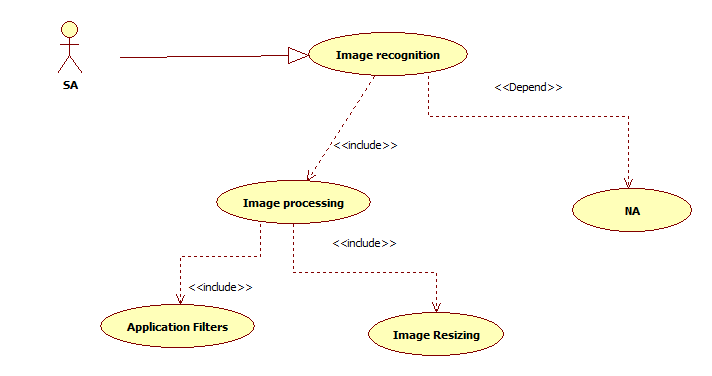
function

* Step 5: Retrieving the edges of an image
* Step 6: Preparing a Mask Image The lightened color image that we mask with edges at the end produces a cartoon image. We use a bilateral Filter which removes the noise.
* Step 7: Giving a Cartoon Effect
* This will be done using MASKING. We perform bitwise and on two
* images to mask them and by QR Code Scanner we can scan that image

**STAGES TO CARTOONIFY THE IMAGE**



USE CASE DIAGRAM



Figure

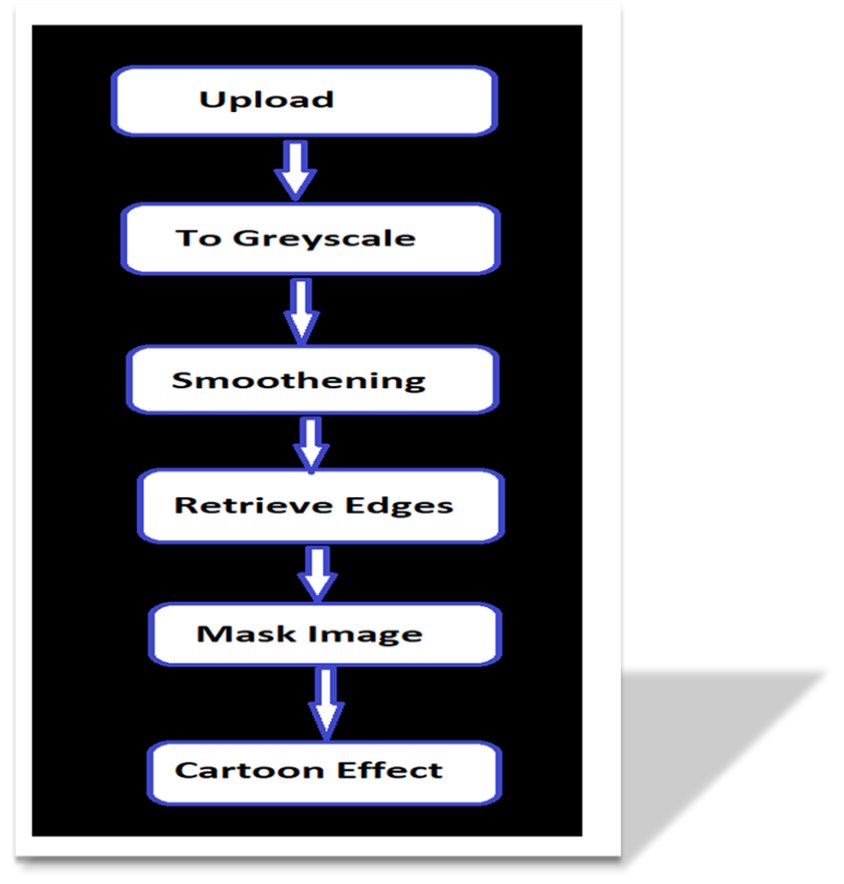


**ORIGINAL IMAGE**

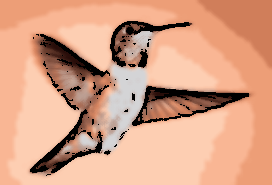


**CARTOONIFY IMAGE**

FLOW CHART

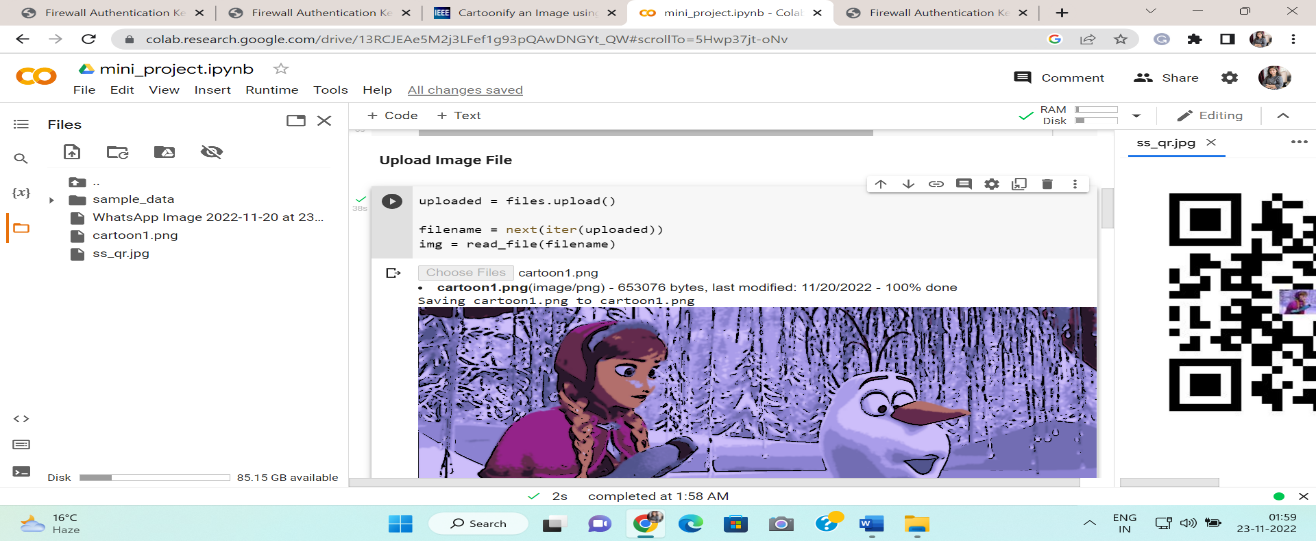
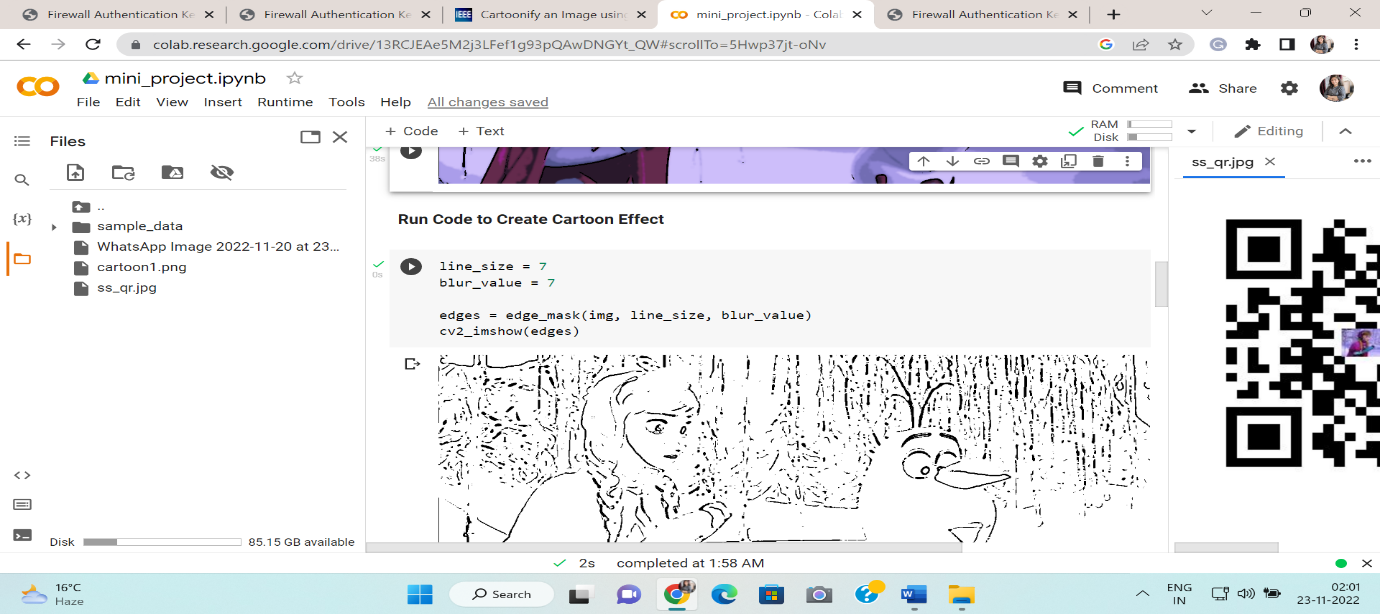
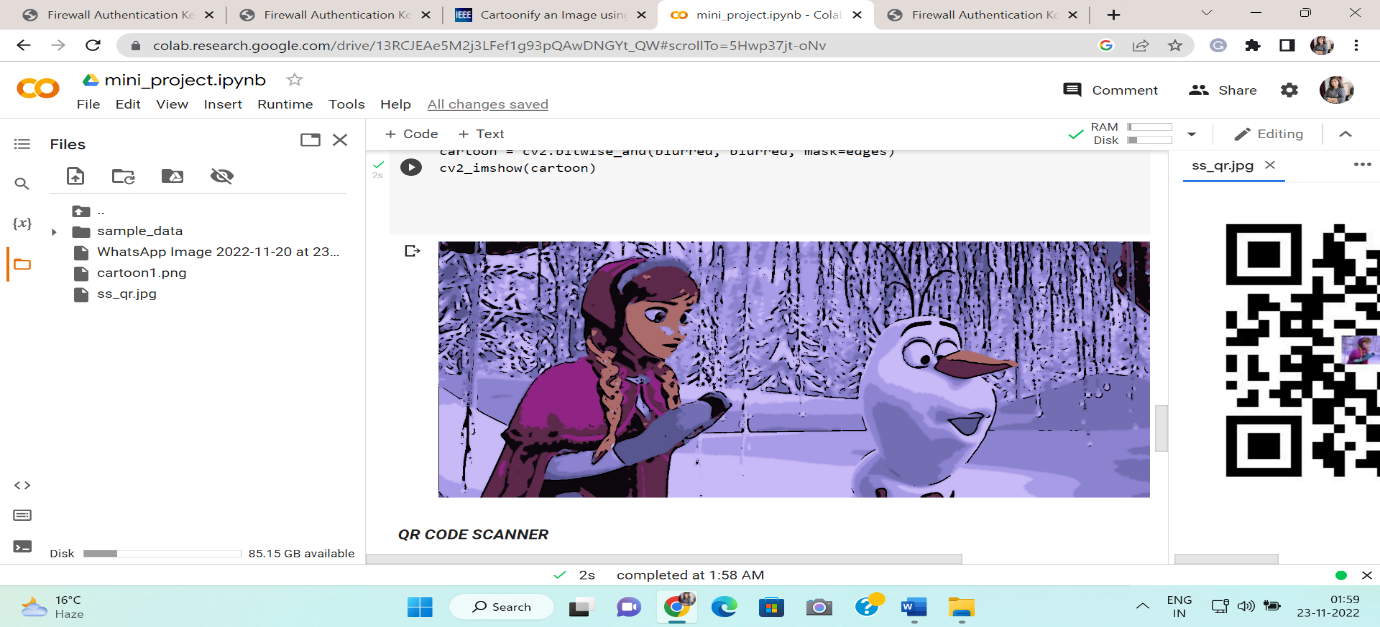


**FLOW CHART OF THE PROCESS**

SOME EXAMPLES OF THE PROJECT

**Figure 1**

**Figure 3**



**FINAL CARTOONIFY IMAGE**

**PRE PROCESSING EDGES OF THE IMAGE**

**IMAGE UPLOADED**

## 

**SKETCH IMAGE**

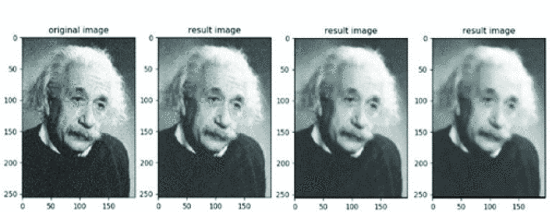
**QR SCANNER**

**QR SCANNER OF**

## **Experimental Analysis**

For image processing filtering the image is an important process that is need to be done. It is done for several things such as, blur removal, edge detection etc. For filtering two types of algorithms are used i.e., Linear and Non-Linear algorithm.

### Gaussian Filter

Median Filter is a non-linear algorithm which is used to remove noise and increases the edge detection of the resultant image. Median filters are used to remove the salt and pepper noise. For the kernel size positive odd integer is used

## 

**Intensity graph**

## describes the distribution of the dark and bright images can be seen all over the black and white images that are choose to convert image into cartoon image. Here the graph indicates that the distribution as the greyscale images that is converted during the greyscale image.

hence when converted into the cartoon image to get the idea of the intensity distribution over the image and its tranquility. After uploading the image or photo that is uploaded, click on cartoon image and it shows the process how an original image is converted into the cartooned image

CODE OF THE PROJECT

**Importing libraries**

import cv2

import numpy as np

from google.colab.patches import cv2\_imshow # use for displaying image

from google.colab import files

def read\_file(filename): # defining the function with its name

  img = cv2.imread(filename)

  cv2\_imshow(img)

  return img

def color\_quantization(img, k):

  data = np.float32(img).reshape((-1, 3))

  criteria = (cv2.TERM\_CRITERIA\_EPS + cv2.TERM\_CRITERIA\_MAX\_ITER, 20, 0.001)

  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

**UPLOAD THE IMAGE**

uploaded = files.upload()

filename = next(iter(uploaded))

**SKETCH IMAGE**

image1=cv2.resize(img,(380,380))

grey\_filter = cv2.cvtColor(image1,cv2.COLOR\_BGR2GRAY)

invert = cv2.bitwise\_not(grey\_filter)

blur = cv2.GaussianBlur(invert,(21,21),0)

invertblur=cv2.bitwise\_not(blur)

sketch\_filter = cv2.divide(grey\_filter,invertblur,scale=256.0)

cv2\_imshow(sketch\_filter)

**CODE FOR CREATING CARTOON**

img = read\_file(filename)

line\_size = 7

blur\_value = 7

edges = edge\_mask(img, line\_size, blur\_value)

cv2\_imshow(edges)

blurred = cv2.bilateralFilter(img, d=7, sigmaColor=100,sigmaSpace=100)

cv2\_imshow(blurred)

cartoon = cv2.bitwise\_and(blurred, blurred, mask=edges)

cv2\_imshow(cartoon)

**QR SCANNER CODE**

!pip install qrcode

import qrcode

from PIL import Image

logo = Image.open('/content/WhatsApp Image 2022-11-20 at 23.48.48.jpg')

basewidth =75

wpercent = (basewidth/float(logo.size[0]))

hsize = int((float(logo.size[1])\*float(wpercent)))

logo = logo.resize((basewidth,hsize),Image.ANTIALIAS)

qr\_big=qrcode.QRCode(error\_correction=qrcode.constants.ERROR\_CORRECT\_H)

qr\_big.add\_data('https://ibb.co/ggkZ0F7')

qr\_big.make()

img\_qr\_big=qr\_big.make\_image(fill\_color='black',back\_color='white').convert('RGB')

pos=((img\_qr\_big.size[0]-logo.size[0])//2,(img\_qr\_big.size[1]-logo.size[1])//2)

img\_qr\_big.paste(logo,pos)

img\_qr\_big.save('ss\_qr.jpg')

**MOTIVATION**

The idea behind choosing this project was to create a cartoon from a picture. This project includes areas of vision (object recognition) and cartoonification. This was thought to be a step in helping the children to better understand and appreciate cartoons. Cartoons seem to be very familiar to children so they can relate to them.

**Future Prospects**

Currently, many systems are facing issues with face cartoonization. This can be improved by providing more facial data with different perspectives to the model. The resolution of the output also needs to be increased. In the future, we can apply these methods on the video to create a cartoon video. Social media is extensively used these days. We keep our online status updated every day, share photos and comments follow our friends’ news. To have a nice profile is a matter of prestige. You can use a photo of your own in a profile image, create an amusing avatar or turn your photo into a cartoon. With a pool of web applications available online, image conversion to a cartoon takes a few clicks.

**CONCLUSION**

In future there will be a project to convert these images into a high definitive image that would help in future, where tried to use loss function to make sure that image is a high definitive but still failed. Next time project will be to convert it into 4K or at least in HD image. Now, Project successfully advanced Image Cartoonifier with OpenCV in Python.

**BIBLOGRAPHY**

## **REFERENCES:**

* <https://realpython.com/tutorials/advanced/>
* <https://data-flair.training/blogs/cartoonify-image-opencv-python/>
* <https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/>
* <https://colab.research.google.com/> (platform used)

**BOOKS:**

* <https://www.javatpoint.com/digital-image-processing-tutorial>
* Deep learning with python and OpenCV by Viral Thakar

## **Online GitHub Repository:**

<https://github.com/ManaliSahu-17/Cartoonify-Mini-Project>

## **Faculty Guidelines:**

Mrs. Ruchi Talwar,(Assistant Professor at GLA University)

