Cartooning of an Image Using Python

Fundamentals of Artificial intelligence
CSCE 5210

Group 22:
Team Members:
Anvesh Chinta
Sai Krishna Reddy Beluri
Srikanth Reddy Guntaka

Abstract:

This project paper describes the process of converting the image which is received as input, to a cartoon-like image with the help of reference images. Cartoon images have a few colour variants with a dark border for the edges. The main aim is to develop a system that gives a Cartoon view of the real input image. It takes a lot of time and space to create a cartoon effect. Currently, the cartoon-effect solutions available are complicated. Other solutions involve performing certain tasks by the user, while others require installing complex software like Photoshop. Toony-Photos is an existing website to perform such a task, but it is difficult to use as the user has to mark down points & lines on the image to apply effects, which isn't very user-friendly. Also, the options are limited. In this regard,

there is a great need for a site that is user-friendly and which is capable of imposing visual effects on images. Bilateral filter and edge detection are the two mechanism we have used in the process of the real image, in which the edges of the real image is identified by the system and the real image is enhanced to a cartoon view image.

Introduction:

The process of converting the image into cartoon-like view is similar to the method of image processing. Exaggerating the real image as the cartoon image is really difficult, so only a cartoon specialist does that. Usually, only cartoon specialists make cartoons. with current sophisticated technology and more libraries in python, we can easily create it using our computers. In the currently available model, the cartoon exaggerates the target. Cartoons have this feature, and that is what makes their cartoons. It is very difficult to convey to every user the exaggeration in the cartoon. A study that creates cartoon-like images with a computer has been developed to help novices easily create cartoons.

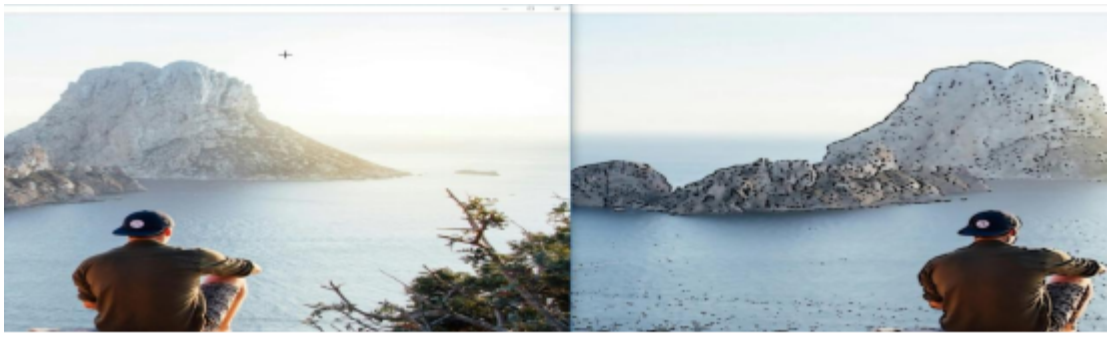
1. Problem Specification:

With current latest and greatest technology, we can use machines to create cartoon-like images in a very short time this is the main advantage of this program but However, most cartoon rendering systems cannot give their results in various ways, since their results are generated according to a fixed algorithm. There are also other cartoon rendering methods that use textures or use user interaction to produce different results. Unfortunately, these methods are not intuitive. Few image processing techniques are performed on the real image, to eliminate functional data and traffic in the image. The processed image is then converted to cartoon-like images by extracting the edges of the image and then applying the median blur of the bilateral filtration method. Hence they turn out to be not that effective. For example below is an image that is not processed well and it doesn't feel like cartooning.

1.1 Data-Set:

In this model, no large amounts of data are used. The picture which is captured by the camera (or) the real world image is sent as input to the model, where the processing is done on the image and the cartoonist view of the image is returned.

The image can be any image that is captured from the real world. The left picture is the real image the right picture is the processed image that looks like a cartoon.



1.2 Problem Analysis:

As we already identified the problem we need to analyse the root cause and try to improve the system from there. The idea was to create cartoons using reference images. By deforming the target to the form of the reference image and adding cartoons to it, the resulting image is obtained. Every user can use it easily. A user can generate a variety of result images by adjusting the intensity of the deformation and cartooning. One of its limitations is that the reference image has to be pre-defined and provided with the feature point model.

2. Design and Milestone:

As we know what to design in this model from performing the problem analysis. We first need few essential parameters to perform this task like with min requirement system as listed below

Minimum Hardware Requirements:

System : Pentium IV 2.4 GHz

Hard Disk : 40 GB

Ram : 512 Mb

Minimum Software Requirements:

Operating system : Windows 7 and above

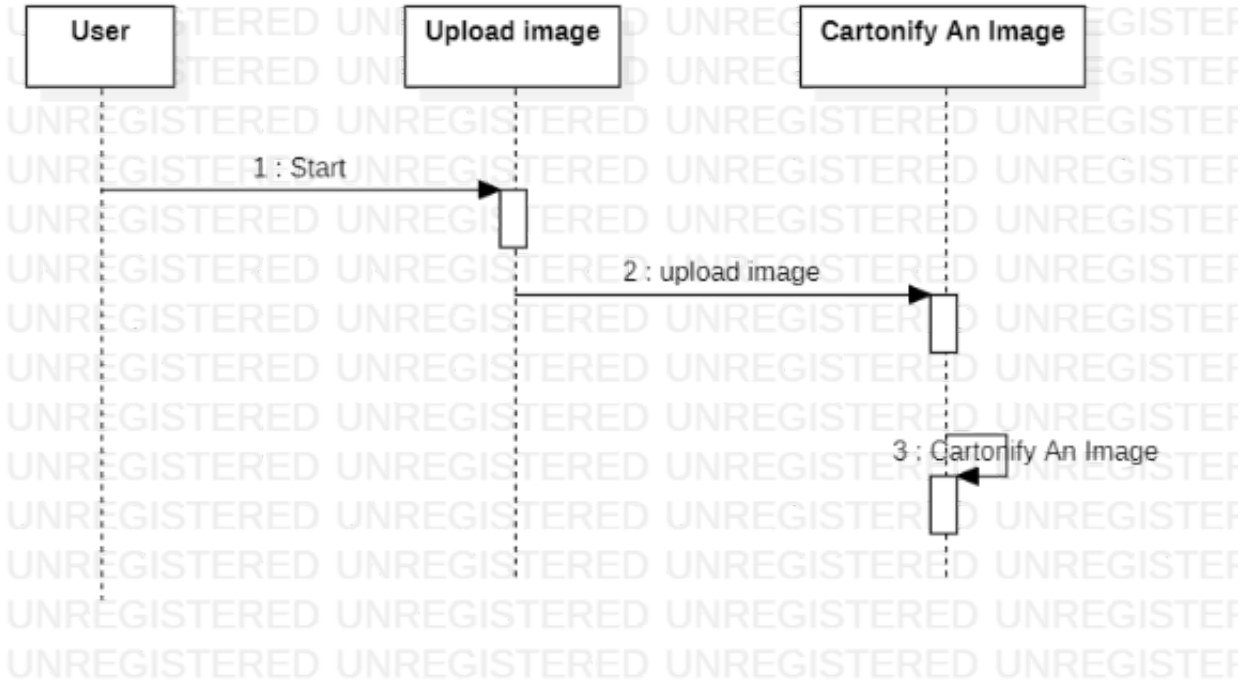
Coding Language : python 3.0 and above

As per our model we just need very few modules to perform the operation of cartoon-like images. Below listed are the required modules.

Modules:

- Upload :In this module user upload image for folder.
- Cartonify An Image: In this module user convert image into cartoon Image.

Below is the sequence diagram designed for this model



As shown in the above sequence diagram the model is simple, flexible and user friendly with fewer components. To define the sequence user starts the process by uploading the image of his choice for the task and by clicking cartooning the image option model will act on it to give output.

2.1 Proposed System:

This model at first downscales the image by applying a bilateral filter to get a cartoon flavour to the real image. A bilateral filter is a filter that processes the image by homogenizing the colour of the image by preserving the edges.

This technique returns the blurred image of the real image which is later converted to greyscale.

The next step is to identify the edges of the images using a technique called Edge Detection. Edge Detection is the main asset of the model which analysis the whole image and identifies the areas which have high contrasts.

2.2 Data Processing:

In this project model, the model accepts the picture and processes the picture by downscaling and extracting the edges, converting it to a grey image and then by applying the bilateral filtering technique image to cartooning image.

Since we are implementing this model in python All below-listed modules such as

pip install NumPy

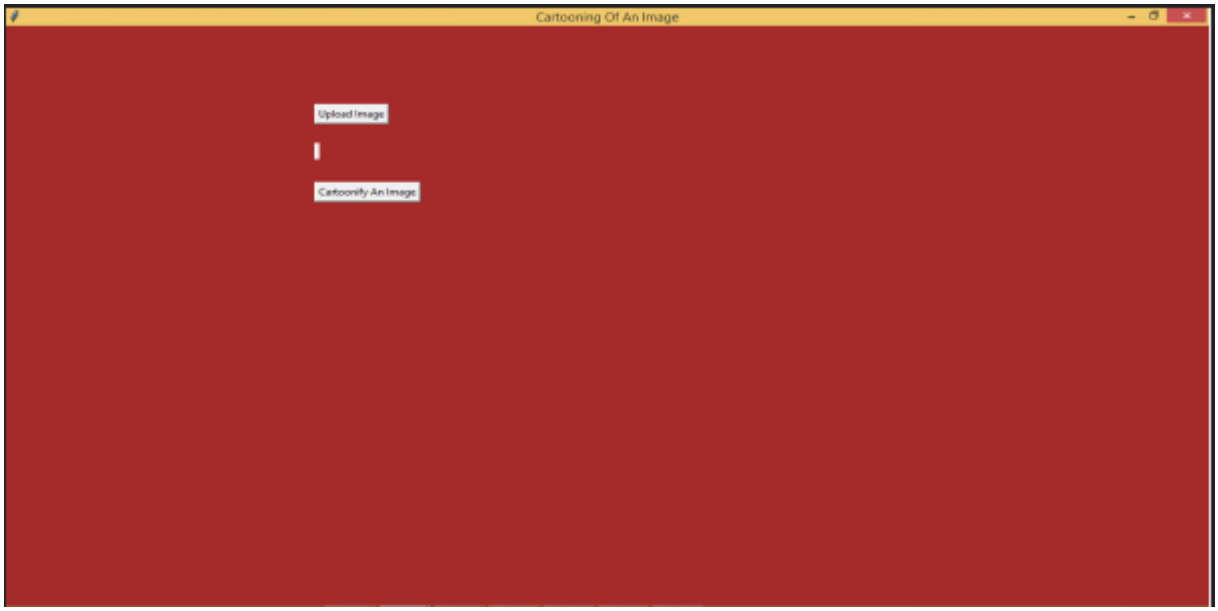
pip install scipy

pip install OpenCV-python

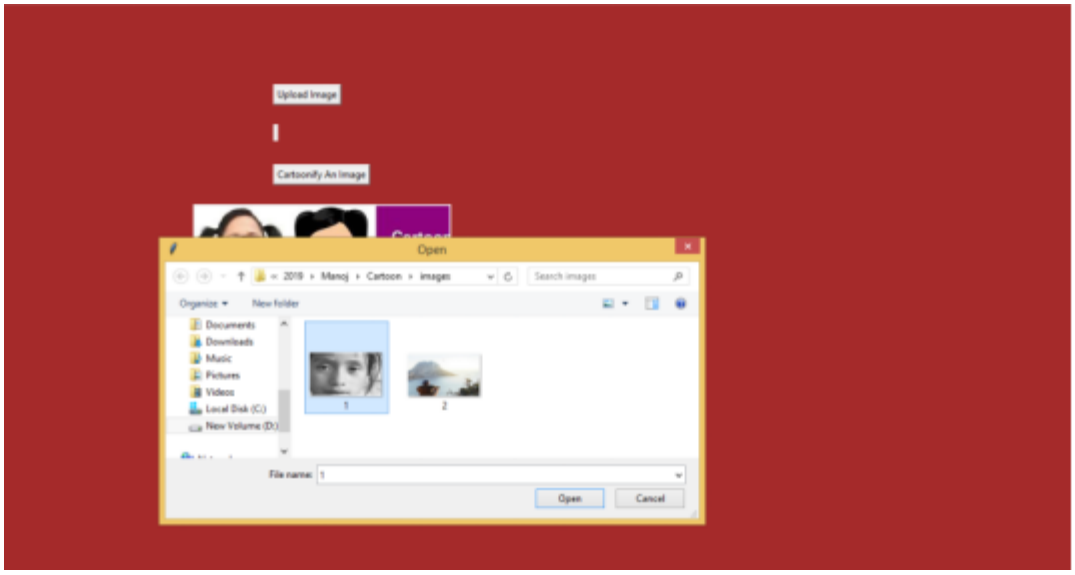
pip install pillow

Are imported for the program to execute.

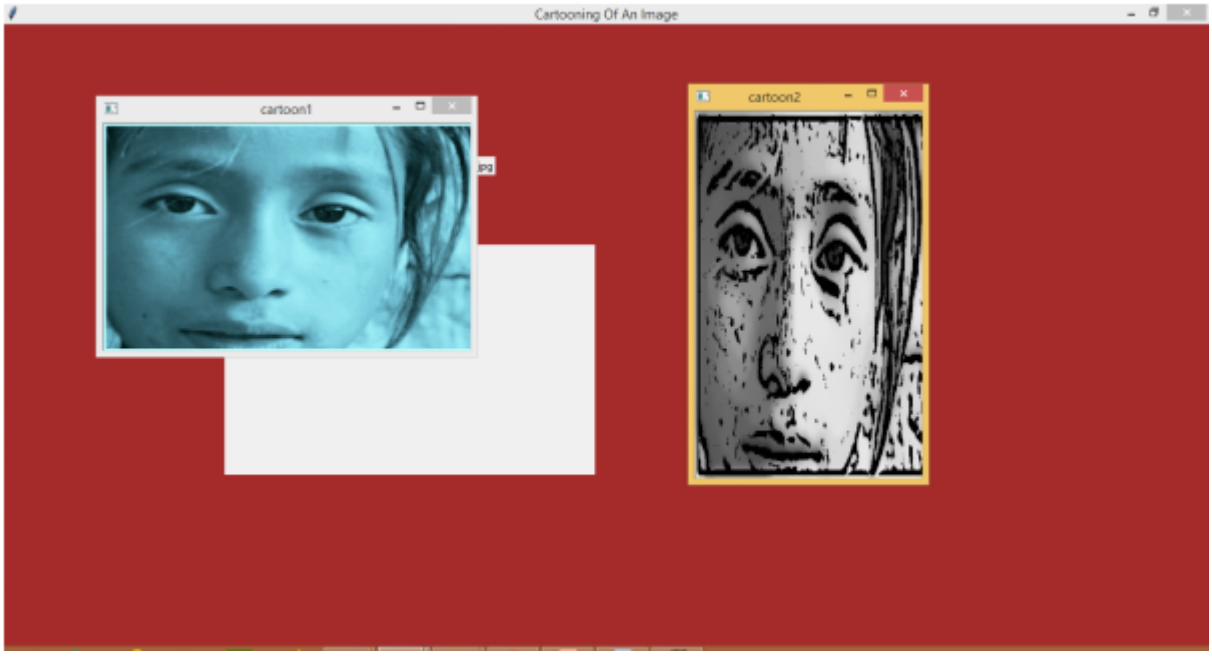
When the model is launched, this kind of interface can be observed



The image which has to be converted into a cartoon should be uploaded first by clicking the upload image button on the screen.



After uploading the real image, by clicking the cartoonify the image button. The image will be converted a cartoon.



The Above image can be considered a preliminary result.

2.3 Experimental Settings:

In the implementation of this model, we deform the target to match the reference image and adding cartooning effect to it, the resulting image is obtained. A user can generate a variety of result images by adjusting the intensity of the deformation and cartooning. By increasing and decreasing contrast, brightness, shadows, exposure and tint we can create so many variations and gets a perfect cartoon effect we need to experiment with a couple of combinations with those settings and choose a few variations of it to get as output. This might give us a better idea of how to interpret the different types of images and give the best outputs.

2.4 Validation methods:

Since this is user driver model there is no specific validation method in order to verify the output. We are checking the best fit settings but manually uploading and checking the outputs and improving the results from the samples.

We are following below exit criteria to validate the best results.

- The operation time should be small and the throughput should be high.
- It should produce timely and accurate results.

Below are the test cases for user requirements:

In Application Home Screen:

Use case ID	Cartooning Of An Image
Use case Name	Home button
Description	Display home page of application
Primary actor	User

Precondition	User must open application
Post condition	Display the Home Page of an application
Frequency of Use case	Many times
Alternative use case	N/A
Use case Diagrams	
Attachments	N/A
Pass criteria	Successfully giving desired output image

Limitations:

- The reference image has to be pre-defined and provided with the feature point model.
- It's hard to pick the perfect settings to produce output since every image is different and has different factors.
- Since it is a computer-generated model for now it is still not perfect when compared to the work of a cartoonist.

Future Work:

The pictures which are returned are cartoon view but are low in quality with fewer colour variants. In future this model can be developed to return the high-quality picture with the 3D feature, even 3D motions can also be added as the extended version for this model.

References:

[1] P. Barla, J. Thollot and L. Markosian, “X-Toon: an extended toon shader”, Proc. 4th interna-tional symposium on Non-Photorealistic animation and rendering, Annecy, pp.127-132, 2006.

[2] https://projectgurukul.org/cartooning-image-opencv-python/

[3] https://www.geeksforgeeks.org/cartooning-an-image-using-opencv-python/

Various other documents and sites on the internet