IMAGE TO CARTOON PYTHON OPEN CV USING MACHINE LEARNING

V. SARALA, DADI USHA,
1.PG STUDENT, D.N.R. COLLEGE, P.G. COURSES (AUTONOMOUS), BHIMAVARAM-534202.
Email id:- ushadadi8@gmail.com
1.Assistant Professor in DEPARTMENT OF MASTER OF COMPUTER SCIENCE, BHIMAVARAM-534202.
Email id:- vedalasarala21@gmail.com

ABSTRACT

This paper represents different techniques of converting image to cartoon. Using any one of below mentioned techniques it is possible to convert all types of captured images to cartoon such as images of person, mountains, trees, flora and fauna etc. There are several other techniques for image to cartoon conversion such as using photoshop, adobe illustrator, windows MAC, paint. Net and much more.

The idea of the paper is based on designated snapshots and videos which are converted to an art form such, as painting. With the help of GAN, it is possible to convert video as well to its carbonized version and the development of the project shows that our Proposed Idea provides high quality cartooned images and videos.

Cartoon-style pictures can be seen almost everywhere in our daily life. Numerous applications try to deal with cartoon pictures, a data set of cartoon pictures will be valuable for these applications.

1. INTRODUCTION

Cartoons are commonly used in various kinds of applications. As we know cartoon made it requires elegant and fine human artistic skills.

Cartoons have always been very popular among children even adults for many years. Cartoon-style pictures can be seen almost every where in our daily life. More and more applications try to deal with this style of pictures, so building a cartoon-style dataset will be valuable. Such a dataset can serve as a training or evaluation bench mark for applications, like semantic classification systems, skeleton extraction methods, cartoon picture modeling systems, etc. It will be beneficial to, various cartoon image applications, like Cartoon-face maker, Art-teaching apps, etc. As a result, we construct a basic cartoon dataset, and build an efficient semantic classification neural network for it.

Constructing a cartoon image data set will be faced with challenges. Cartoon images are not the snap shots of real-world objects, so it is difficult to make use of existing resources to generate cartoon pictures. Although a number of cartoon images from the Internet are available, most of them have the similar cartoon-style: one big color block connected with an other. You can use a photo of your own in a profile image, create an am using avatar or turn your photo into a cartoon. With a pool of web applications available online; an image conversion to cartoon takes few clicks.

A couple, of years ago, the styling of images consists of a particular domain named “on-photorealistic rendering”. The Traditional algorithm was developed on the base of the domain for the styling of images and they were successful in styling an images by adding designs, texture, effects, etc.With the help of the algorithm, much software was developed to convert

real images (snapshot) into cartoon images some of the methods failed while some of the methods gave results but didn't satisfy all the requirements. Moreover, cartoon images are by compared to real life images.

2.LITERATURESURVEYANDRELATEDWORK

2.1 Image Processing

Step1: Importing the required modules . We will import the

following modules:

CV2:Imported to use Open CV for image processing

Easy Gui: Imported to open a file box. It allows us to select any file from our system.

Num Py: Images are stored and processed as numbers. These are taken as arrays. We use Num Py to deal with arrays.

Image Io:Used to read the file which is chosen by file box using a path.

Matplot lib: This library used for visualization and plotting. Thus, it is imported to form the plot of images.

OS: For OS interaction. Here, to read the path and save images to that path.

Step2: Building a File Box to choose a particular file

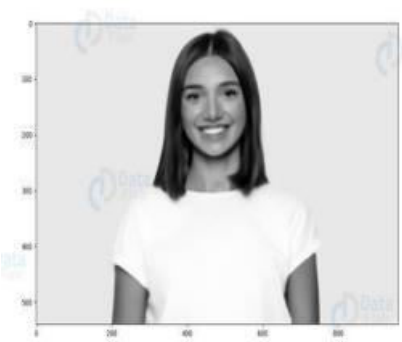
In this step, we will build them a in window of our application where the buttons, labels, and images will reside.

Step3:

Beginning with image transformations: To convert an image to a cartoon, multiple transformation done. Firstly, an image is converted to a Gray scale image. Yes, similar to the old day's pictures.! Then, the Grayscale image is smoothened, and we try to extract the edges in the image. Finally, we form a color image and mask it with edges. This creates a beautiful cartoon image with edges and lightened color of the original image.

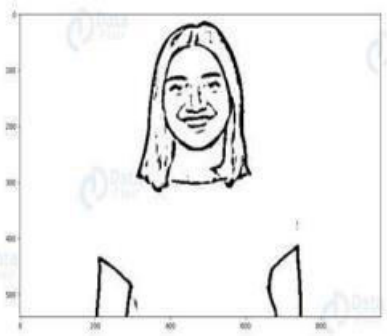
Step4:Transforming an image to grayscale

Step5:Smoothening a gray scale image and simply apply a blur effect.



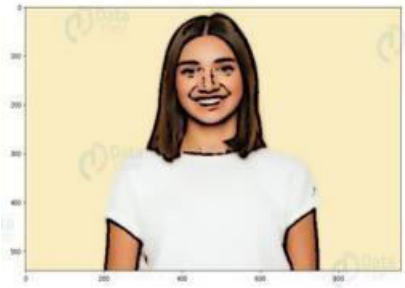
Retrieving the edges of an image

In this step, we will work on the first specialty. Here, we will try to retrieve the edges and high light them.

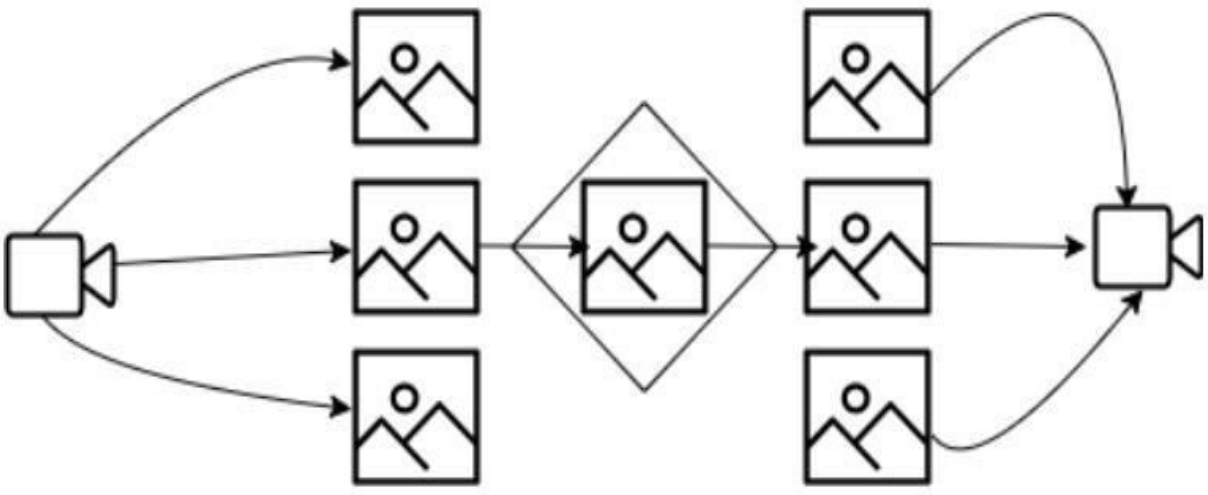


Step7:Preparing a Mask Image

Step8:Giving a Cartoon Effect



tep9:Plotting all the transitions together



3. EXISTING SYSTEM

1.OpenCV:

OpenCV is a powerful open-source computer vision library that provides a wide range of tools for image processing and manipulation. You can use OpenCV for tasks like image preprocessing, edge detection, and image filtering, which are essential for image-to-cartoon conversion.

2.Machine Learning Models:

There are several machine learning models and architectures that have been applied to image-to-cartoon conversion. Some popular options include:

CycleGAN:

A generative adversarial network (GAN) architecture that can be used for image translation tasks.

U-Net:

A convolutional neural network (CNN) architecture often used for image segmentation and translation.

Pix2Pix:

A conditional GAN model designed for paired image-to-image translation tasks.

3.Datasets:

You'll need a dataset of paired images, consisting of real photos and their corresponding cartoon versions, for training your machine learning model. Datasets like the "CMP Facade Database" or custom datasets created by collecting image pairs can be used.

4.Python Libraries:

Python is a popular language for machine learning and computer vision tasks. Libraries such as TensorFlow, PyTorch, and Keras can be used for building and training machine learning models.

5.Pre-trained Models:

You can leverage pre-trained models for various computer vision tasks and fine-tune them for image-to-cartoon conversion. For instance, using a pre-trained VGG or ResNet model as a feature extractor can be beneficial.

6.Image Processing Techniques:

OpenCV offers a range of image processing techniques like bilateral filtering, edge detection, and color manipulation that can be applied to enhance the cartoonization effect.

7.Research Papers and Tutorials:

There are numerous research papers and online tutorials available that describe the implementation of image-to-cartoon conversion using machine learning. These resources often provide code examples and insights into the latest techniques.

8.Community and Forums:

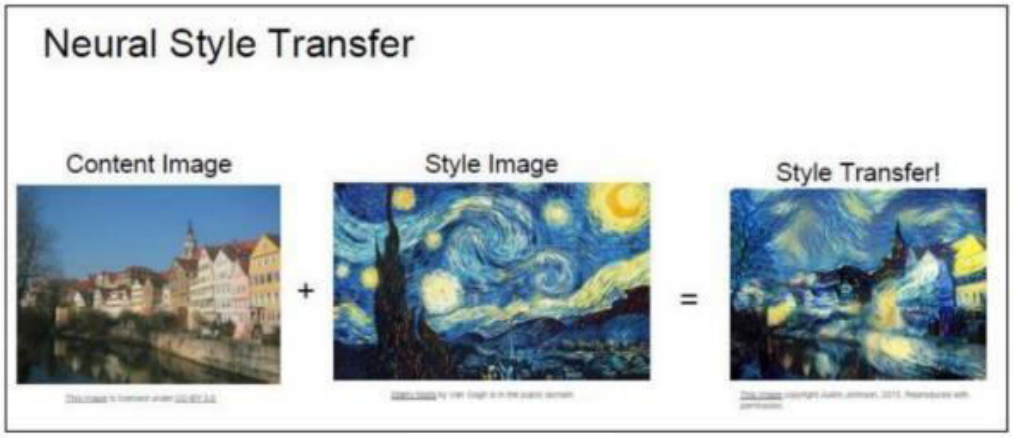
Online communities and forums like Stack Overflow, Git Hub, and specialized machine learning forums are excellent places to seek help and collaborate with others working on similar projects.

4. PROPOSED SYSTEM

We propose neural style transfer which is a machine learning algorithm, which involves two images, first is the input image from the user and second is the style image which is used to apply the style on the input image. Following are the examples of images generated using neural style transfer.

We propose to create a website, which consists of image upload functionality using which the 1user can upload his image, the uploaded image is then processed by server using Neural style transfer algorithm and the resulting image is presented to the user on the website. Which then user can download & share. Neural fast style transfer is used by Apps such as https://deepart.io, Prisma, Artistoetc. We decided to choose this approach over traditional image filters (e.g., using image filters such as median & bilateral filters to posterize an Image) as Neural fast style transfer is quite new and challenging technique which uses machine learning & image processing to produce various styled i1mages based variety of Input & style images.The algorithm can be implemented in Python/JavaScript/Lua to perform neural style transfer. We will use Python to implement the back end and the frontend of the website will be in HTML,CSS&JS.

Basically, in Neural Style Transfer we have two images- style and content. We need to copy the style from the style image and, apply it to the content image. By, style we basically mean, the patterns, the brushstrokes, etc. we will provide a set of style images which a user can use to apply different kinds of Cartoon like effects this image.



5. IMPLEMENTATION

MODULES

1.OpenCV (Open Source Computer Vision Library):

- OpenCV is the core library for image processing and computer vision tasks. It provides a wide range of functions for reading, manipulating, and saving images, as well as advanced image processing capabilities.

2. **NumPy**:

- NumPy is a fundamental library for numerical operations in Python. It's often used in conjunction with OpenCV for handling image data efficiently, as images can be represented as NumPy arrays.

3. **Matplotlib or Pillow (PIL)**:

- Matplotlib is a library commonly used for creating plots and visualizations in Python. It can be used to display images and results.
- Pillow (PIL) is an alternative library specifically designed for image processing tasks. It's useful for opening, manipulating, and saving various image formats.

4. **TensorFlow, PyTorch, or Keras** (Machine Learning Libraries):

- You'll need one of these deep learning libraries to implement and train machine learning models for image-to-cartoon conversion. These libraries provide pre-built neural network layers and tools for model development and training.

5. **Scikit-learn** (Optional):

- Scikit-learn can be used for various pre-processing tasks, such as feature scaling, dimensionality reduction, or data splitting if your image-to-cartoon application involves more complex machine learning workflows.

6. **Tkinter or PyQt (GUI Libraries)** (Optional):

- If you want to create a graphical user interface (GUI) for your application, you can use Tkinter (Python's standard GUI library) or PyQt (a popular third-party GUI library).
- **Pandas** (Optional):
- Pandas is helpful for data manipulation and analysis. You may use it to organize and manage datasets if your project requires custom data preparation.

7. **Jupyter Notebook** (Optional):

- Jupyter Notebook is a popular interactive environment for data analysis and prototyping. It's useful for experimenting with image processing and machine learning algorithms.

8. **Python Imaging Library (PIL)**:

- PIL is a powerful image processing library that can be used alongside OpenCV for specific image manipulation tasks.

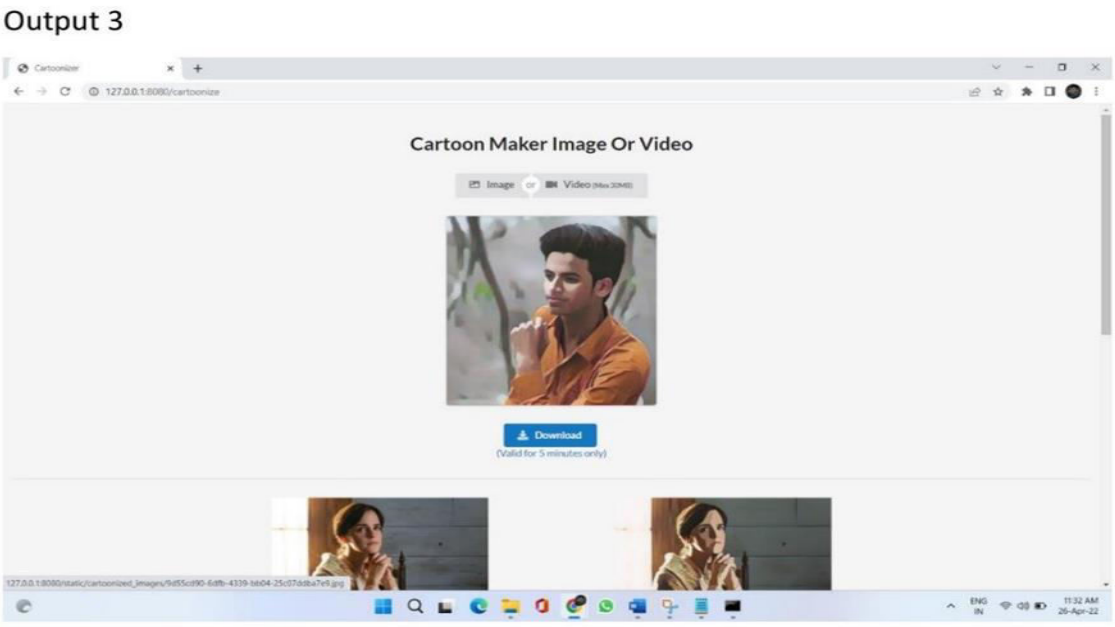
9. **Requests (for web-based applications)**:

- If your image-to-cartoon application interacts with web services or APIs for data retrieval or other purposes, the **requests** library is useful for making HTTP requests.

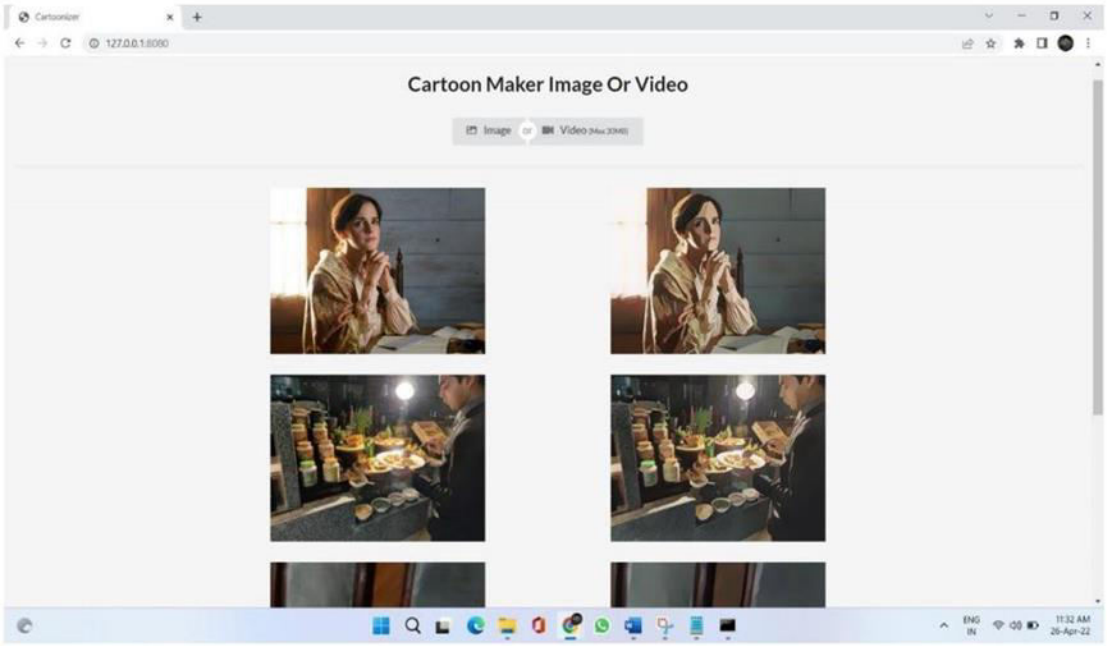
6. RESULTSANDDISCUSSION SCREEN SHOTS

Screen Shots:

output Screens:



Output 2



7.CONCLUSION AND FUTURE WORK

CONCLUSION:

Thus, we have shown that how image can be converted to cartoon. We also stated the examples on how image is converted to cartoon. Hardware and software requirements of image to cartoon conversion are also shown in this paper. The systematic working of image to cartoon conversion and respective algorithm and formulae is how with neat diagram in this paper. Also, we have stated challenges and problems one can face while carton the captured image. In this paper we have also discussed need and scope of carbonising the content image.

In this paper with the help of Cartoon GAN, where GAN stands for Generative Adversarial Network isused to transform images (snapshots) to the finest cartooned image (animated image). With the help of the loss function and its two types named as Adversarial loss and Content Loss, we got a 37lexible as well as a clearedge defined images. Also, with the help of CV2 which is known as Computer vision, we have transformed onto animation (cartoonizedvideo).

FUTURE WORK

The project has successfully demonstrated the conversion of images into cartoon-style images using Cartoon GAN. Additionally, video clips were transformed into animation clips with the help of the Python library cv2. In future work, the focus will be on improving the generation of high-definition (HD) portrait images, despite using loss functions that did not yield the desired results initially. The project also aims to enhance the video conversion process to achieve HD or even 4K quality videos, which would be more beneficial to users.

The proposed system will offer users a selection of pre-trained style images to choose from. Based on the chosen style and the

content image provided by the user, the program will generate resulting images with a cartoon-like effect. The implementation is based on a combination of various techniques, including Gatys' A Neural Algorithm of Artistic Style, Johnson's Perceptual Losses for Real-Time Style Transfer and Super-Resolution, and Ulyanov's Instance Normalization.

This future work aims to improve the quality and flexibility of image and video transformations, offering users more options and higher-quality results.

8. REFERENCE

1.**OpenCV Documentation**:

- The official OpenCV documentation provides a wealth of information on image processing techniques and functions. You can refer to it for details on specific OpenCV functions used in your code.
- Website: OpenCV Documentation

2. **Tutorials on Cartoonization**:

- Various online tutorials and articles cover the image-to-cartoon conversion process using OpenCV. These tutorials often provide step-by-step guidance on implementing the technique.
- Example: Image Cartoonizer with OpenCV in Python

3. **Image Processing Books**:

- Books on image processing and computer vision can be valuable resources. They often cover the underlying concepts and algorithms used in image manipulation.
- Example Book: "OpenCV 4 Computer Vision with Python Recipes" by Joseph Howse

4. **Academic Papers**:

- Some image processing and computer vision research papers may discuss related techniques and algorithms. While they may not directly relate to your code, they can offer insight into the field.
- Example Paper: "Bilateral Filtering for Gray and Color Images" by C. Tomasi and R. Manduchi

5. **Online Forums and Communities**:

- Websites like Stack Overflow and the OpenCV forum are excellent places to ask questions and learn from others who have worked on similar projects.
- Example: OpenCV Forum

6. **YouTube Tutorials**:

- YouTube offers many video tutorials on image processing with OpenCV, which can be helpful for visual learners.