Cartoonifying Image In Real Time

Introduction

- *Cartoonifying an image is a fascinating process that involves transforming a regular photograph into a cartoon-like representation.*
- *Cartoonifying an image using Python ,OpenCV and Numpy is a fascinating and creative image processing task that can transform a regular photograph into a stylised cartoon-like representation.*
- *This technique is often used for artistic purposes, graphic design, and even in creating cartoon avatars for social media profiles.*

What is Cartoonisation?

- *Cartoonisation, also known as cartooning, is a process of transforming a regular image or photograph into a cartoon-like representation*
- *Cartoonisation is typically done for artistic or creative purposes, as well as for various applications such as caricatures, digital art etc.*
- *The process of cartoonisation often involves techniques like edge detection, smoothing, and colour simplification to achieve the desired cartoon-like effect.*
- *There are various software tools and algorithms available for cartoonising images, including both manual and automated approaches.*

Overview of Tools

- *Python ,OpenCV and Numpy can be used to cartoonify the image.*
- *Here the tool used in cartoonifying image:*
- **Image Loading:**
- OpenCV(cv2.videocapture()): *OpenCV is a powerful library for image processing tasks in python. Using this function ,we are getting the real time video.*
- OpenCV(img.read()): *The video we have captured will save as ing variable. Use the read() function to load the input image that you want to cartoonify.*

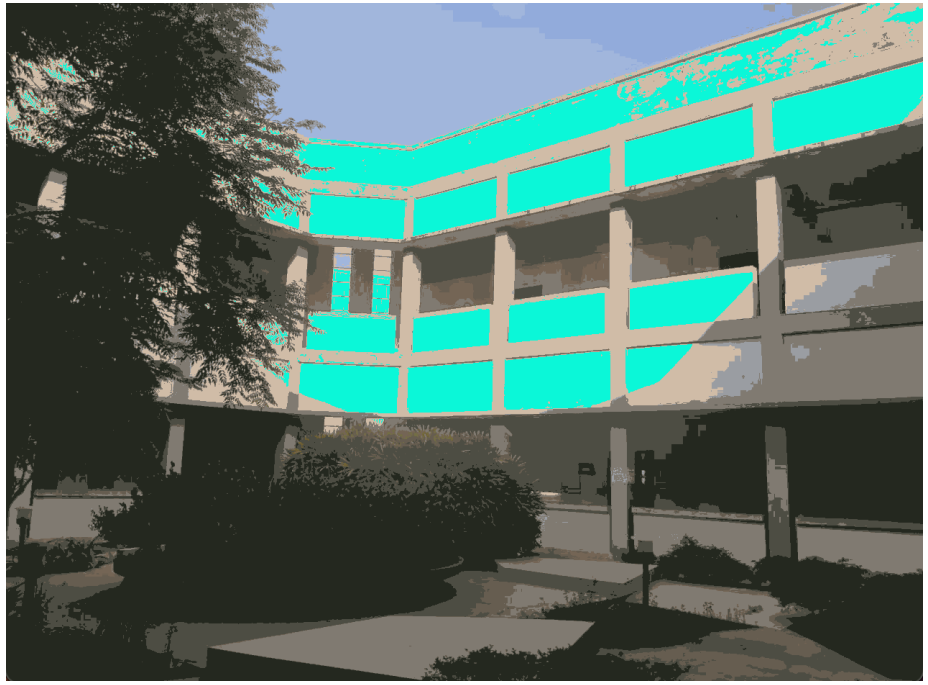
- **Image processing:**

- Resahpe of image(img.reshape()): *We are reshaping the image, It means that we are going to convert the image into numbers. These numbers help us to determine colour use in image and also help us to select centres*
- *Convert into float(np.float32()): Using this function, we are converting the numbers get from the "img.read()" into decimal values*
- **Define criteria**:

- OpenCV(cv2.TERM_CRITERIA_EPS()) : *the optimisation process will stop when the change in the objective function (or other relevant metric) becomes smaller than a certain threshold (epsilon).*
- OpenCV(cv2.TERM_CRITERIA_MAX_ITER()) : *It is a constant used in OpenCV for specifying the termination criteria for iterative optimisation algorithms.*

- **Apply kmeans:**
- Number of Clusters(cv2.kmeans()) : *The cv2.kmeans function in OpenCV is used for performing the k-means clustering algorithm on a set of data points. K-means clustering is an unsupervised machine learning technique that is often used for grouping data points into 'k' distinct clusters based on their similarity.*
- Grouping Data into Clusters : *Given a set of data points, the cv2.kmeans function groups these data points into 'k' clusters, where 'k' is the number of clusters you specify as a parameter.*
- **Making Original Image :**
- Convert into Decimal(np.uint8()) : *It stands for "NumPy unsigned 8-bit integer." It's a specific data type used to represent whole numbers in the range from 0 to 255.*
- Reshaping the Image(res.reshape()): *This method is used to change the shape or dimensions of a NumPy array.* Finally ,*The cartoonify image will be created.*

Demonstration



Demonstration

Real Image

Cartoonised Image

Conclusion

cartoonifying an image using Python and OpenCV is a fascinating and creative process that involves simplifying and stylising a photograph to give it a cartoon-like appearance. By leveraging the powerful capabilities of OpenCV and Python, developers and artists can transform ordinary images into visually appealing and artistic cartoons. Key aspects of cartoonifying an image include grayscale conversion, edge detection, bilateral filtering, combining edges with the smoothed image, colour enhancement, and artistic adjustments.