Cartoonify Realistic Images and Videos Using OPENCV

Bensy Benny
PG Student
Dept. of computer Applications
Amal Jyothi college of Engineering
Kanjirappally, Kottayam
bensybennycd@gmail.com

Ms. Sona Maria Sebastian
Asst. Professor
Department of Computer Applications
Amal Jyothi college of Engineering
Kanjirappally, Kottayam
Sonasebastian@amaljyothi.ac.in

***Abstract*— Nowadays we keep Online Status consistently, share photographs and remarks. To have a pleasant profile we can utilize our own photograph in a profile picture, make an entertaining symbol or transform your photograph into an animation. Therefore this paper representing the technique of converting realistic images and videos into cartoon. The objective of this study is to explore possibility of applying the application of a Generative Adversarial Network (GAN) with using two loss functions, content loss and adversarial loss for getting a sharp and clear image and converting realistic images and videos into cartoonized version with the help of K- means clustering. This article has conducted a comparative study on the performance of models created using deep learning based generative model architecture and K-means clustering. This system also evaluates those parameters on both algorithms. The histogram functionality are processed in python programming language using Convolution block and clustering.**

Keywords:

Deep Learning, Python Programming, Convolutional Neural Network, K- Means Clustering.

I. INTRODUCTION

Machine learning is a growing technology which enables computers to learn automatically from past data using various algorithms. Data is generated utilizing two models, such as the Generator model and Discriminator model, using the Generative Adversarial Network (GAN), a deep learning-based generative model architecture. Additionally, the two models compete with one another to produce predictions that are more accurate. Generative modeling is an unsupervised learning task in machine learning that involves discovering and learning naturally occurring normalities or examples in input data in order to use the model to produce or produce new models. Cartoon visuals are also more sophisticated than real images.

Review reports claim that it takes a lot of effort and space to create a cartoon-like look. It takes a lot of time for the artists to illustrate cartoons in large quantities for any animated data since they must accurately characterize the animation's sketch to provide a decent result. Simple and entertaining photo effects applications are becoming more and more popular. Then, with the aid of the algorithm, a variety of programming was developed to convert real images into animation images. Some of the techniques failed, while others produced results but didn't completely satisfy the requirements.

This study aims to provide a comparative analysis on various applications of Generative Adversarial Network(GAN), convolutional Neural Network and K-Means Clustering for converting real image into animated form.By defining the issue as supervised learning with two models, Generator and Discriminator, the Generative Adversarial Network is utilized to train generative models. These models are used to train new models and categorize them as authentic or phony. These models compete with one another to produce a superior outcome. Python is a programming language that is used to execute these models. It is simpler to implement OpenCV and K-Means Clustering. A clustering algorithm called K-Means Clustering divides n observations into K groupings. It can be used to group items together based on similarities between them.

II. LITERATURE SURVEY

This section reviews some papers related to similar studies and are reviewed. A real image is converted into an animated image with the help of the application developed known as “Cartoon GAN” in less duration of time in [1]. Authors developed a system to convert real images or videos into cartoon by using about 3000 images for training the model from Flickr and training data consist of real world images which were passed through 200 epochs trained to produce the cartoon model with using adversarial function and loss function and the final output is generated with the convolution block with fps 0.5s

Chinmay Joshi, Devendra Jaiswal, Akshata Patil proposed “Technical Paper Presentation on Application of Cartoon like Effects to Actual Images” [2]. Authors

DOI: 10.5281/zenodo.6905304
ISBN: 978-93-5607-317-3@2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

representing different techniques of converting image to cartoon. They used 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 on the input image.

Nikhil Nautiyal, Veernarayan Sinha, Savleen Kaur in "Image Cartoonization" [3] consist of cartoonization of an image using Generative Adversarial Network a machine learning framework that generated new data based on training data and OpenCV library. Also using filters such as medianBlur() , GaussianBlur(), Laplacian(), BilateralFilter() and many more.

Vaishali Sudarshan in "Cartooning an image using OpenCV and Python" [4] using image processing by identifying an object in image, identify the dimensions, no of objects, changing the images to blur effect and such effects are highly appreciated. Their method achieved good accuracy than other approaches.

Different from earlier work on Cartooning Real images and videos, This article propose a easy method to convert into cartoon using Generative Adversarial Network, Convolutional Neural Network and K-means clustering with using OpenCV.

III. MOTIVATION

An animation makes a short, entertaining remark utilizing representations and is many times tracked down 0n TV, in a paper or in a magazine. Watching cartoons can help to develop logic and reasoning ability and a selective attention. Cartooning realistic images and videos can provide an extremely fun view. It is a perfect way to provide our photos in a new style.

Cartoonify uses neural networks to turn your realistic images and videos into a unique cartoon view. Numerous programs were developed to convert real photographs into cartoon views with the aid of the various ways. Some of the strategies succeeded, while others produced results but didn't satisfy all the requirements. Generative Adversarial Network (GAN) is the key tool we use to help us meet all the requirements for turning real photos into cartoon versions. A cartoonized image depicts a scene in unique contrast to how our human vision sees the world in comparison to a camera's focus point.

IV. METHODOLOGY

The purpose of this paper is to convert realistic images and videos into cartoonized view using Generative Adversarial Network (GAN) and K- Means Clustering. In this system we use the pycharm to establish a helpful environment. Then with the use of OpenCV and K-Means Clustering to perform picture handling and cartoony them. This shows how great picture handling can be utilized to do tasks. GAN is a combination of Generative model and Discriminator model. The Generator model makes new occasions of data. The Discriminator is the model utilized for testing the information and contrasting it and the image from the Generator and chooses whether the image is real or fake.

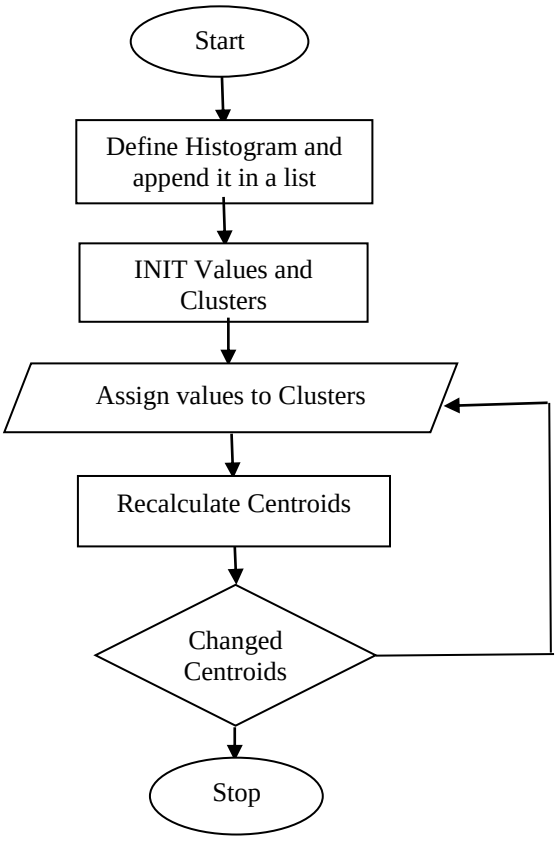

Figure 1: Image conversion using K-Means Clustering.
The steps require to be followed are:

A. Requirements

- Import Packages
- Pycharm
- Images
- Webcam

B. OpenCV

OpenCV is a open-source library for the computer vision, machine learning, and image processing. It can process images and videos to identify objects.

DOI: 10.5281/zenodo.6905304
ISBN: 978-93-5607-317-3@2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

C. Generative Adversarial Network

I've used GAN algorithm in the training phase which have two models Generator and Discriminator. Both are neural networks compete with each other and repeated multiple times to get a better result.

D. K-Means Clustering

K-Means Clustering is an unsupervised learning algorithm that is utilized to tackle the clustering issues in data science or machine learning.It is an iterative calculation isolates the unlabeled dataset into K different clusters so that each dataset has a place just a single gathering that has comparative properties.The variable K represents the number of groups or categories made. The objective is to divide the data into K different clusters and report the area of the focal point of mass for each cluster. Then a new data point can be assigned a cluster based on the center of mass.

K-Means clustering mainly performs two tasks:

1) Find the best value for K center points or centroids by an iterative process.
2) Assign every information highlight its nearest K-center. Those information focuses which are close to the specific K-center, make a group.

It is a centroid based algorithm, where each bunch is related with a centroid. The fundamental point of this calculation is to limit the amount of distances between the data point and their related clusters. Algorithm for this method is shown below as Algorithm.

Algorithm : Converting realistic images and videos using Generative Adversarial Network (GAN) and K-Means Clustering using OpenCV. Input: Giving access to Webcam. Output: We import the cartoon function and call it for each frame.

Step 1: Give access to Webcam using OpenCV.

Step 2: Using Generative Adversarial Network (GAN) we train it's generative model by framing the image.

Step 3: Convert the format of image from RGB to HSV by drawing histogram using histogram function of each and append it in a list.

Step 4: Apply K-Means Clustering.

Step 5: Create a function to update the value of the centroid and find the centroid value for each new cluster

Step 6: Create a new function with the used name, defines the number of centroid values for each channel and calls the above function.

Step 7: Cluster each value to the centroids which reshape the image.

Step 8: Import the cartoon function and call it for each frame of the visual.

V. IMPLEMENTATION

The model building is the main step in the cartoonifying realistic images and videos. While building the model user use the algorithms. The steps involved are:

1. Import the packages that are necessary.

```
import cv2
import scipy
from scipy import stats
import numpy as np
from collections import defaultdict
```

```
import numpy as np
import cv2
from cartoonize import caart
```

2. Load sample image and video and train to recognize it.

```
videoCaptureObject = cv2.VideoCapture(0)

out = cv2.VideoWriter('out.mp4', cv2.VideoWriter_fourcc(*'MP4V'), 24, (720, 1280))
result = True
while(result):
    ret,img = videoCaptureObject.read()
    img=caart(img)
```

3. Creating a function to update the value of centroid using histogram function and append it in a list.

```
def update_c(C,hist):
    while True:
        groups=defaultdict(list)

        for i in range(len(hist)):
            if(hist[i] == 0):
                continue
            d=np.abs(C-i)
            index=np.argmin(d)
            groups[index].append(i)
```

4. Then create a new function and define the number centroid values of each channel and call previous function.

DOI: 10.5281/zenodo.6905304
ISBN: 978-93-5607-317-3@2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

```
        new_C=np.array(C)
        for i,indice in groups.items():
            if(np.sum(hist[indice])==0):
                continue
            new_C[i]=int(np.sum(indice*hist[indice])/np.sum(hist[indice]))

        if(np.sum(new_C-C)==0):
            break
        C=new_C

    return C,groups
```

5. Then, clustering each value and reshape it.

```
def K_histogram(hist):

    alpha=0.001
    N=80
    C=np.array([128])

    while True:
        C,groups=update_c(C,hist)
```

```
for h in hists:
    C.append(K_histogram(h))


output=output.reshape((-1,c))
for i in range(c):
    channel=output[:,i]
    index=np.argmin(np.abs(channel[:, np.newaxis] - C[i]), axis=1)
    output[:,i]=C[i][index]
output=output.reshape((x,y,c))
output=cv2.cvtColor(output, cv2.COLOR_HSV2RGB)
```

6. Finally import the cartoon function and call it for each frame of the video.

```
    img=caart(img)
    cv2.imshow("original",np.array(img))
    out.write(img)
    if(cv2.waitKey(1) & 0xFF==ord('q')):
        break
videoCaptureObject.release()
cv2.destroyAllWindows()
```

VI. RESULT

The result shows that the captured frame in a cartoonized form. Cartoon images play essential roles in our daily existences particularly in entertainment, education and advertisement, that become an increasingly concentrated research in the field of multimedia and computer graphics.
We will start by working on images as input by accessing webcam. Then we train it using Generative Adversarial Network using Generator and Discriminator models. After the use of OpenCV and K-Means Clustering it reshapes the image and display it in a cartoonized form.

VII. CONCLUSION

This paper provides a method for converting realistic images and videos using webcam into a cartoonized view. This is using best features to cartoonify realistic images and videos. Here after accessing the webcam, implementing the solution by processing the data using Generative Adversarial Network(GAN) to train the input using two models, Generator and Discriminator. Then using K-Means clustering defining number of centroids and reshape the image. After reshape import the cartoon function and call it for each frame of the visual. Then we can see a finest cartoon-style images as the output.

In the future work, we would like to improve cartoon stylization for human faces with expecting HD quality. We also plan to add sequential constraints to the training process to extend our method to handling videos.

REFERENCES

[1] Akanksha Apte, Ashwathy Unnikrishnan, Navjeevan Bomble, Prof. Sachin Gavhane “Transformation of Realistic Images and Videos into Cartoon Images and Video using GAN”, International Research Journal of Engineering and Technology(IRJET), vol. 07, no. 01, January 2020

[2] M.A. Jabbar,” Computational intelligence technique for early diagnosis of heart disease”, ICETECH 2015, IEEE, pp1- 6(2015)

[3] Nikhil Nautiyal, Veernarayan Sinha, Savleen Kaur “Image Cartoonization” , International Journal for Modern Trends in Science and Technology, vol. 07, no. 08,May 2021

[4] Vaishali Sudarshan,”Cartooning an Image Using Opencv and Python “, International Journal of Interdisciplinary Innovative Research and Development(IJIIRD), vol. 04, no. 02,2020

DOI: 10.5281/zenodo.6905304
ISBN: 978-93-5607-317-3@2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam