National Institute of Technology, Calicut

Cartooning Of An Image - Report 2
Image Processing

Under the guidance of :

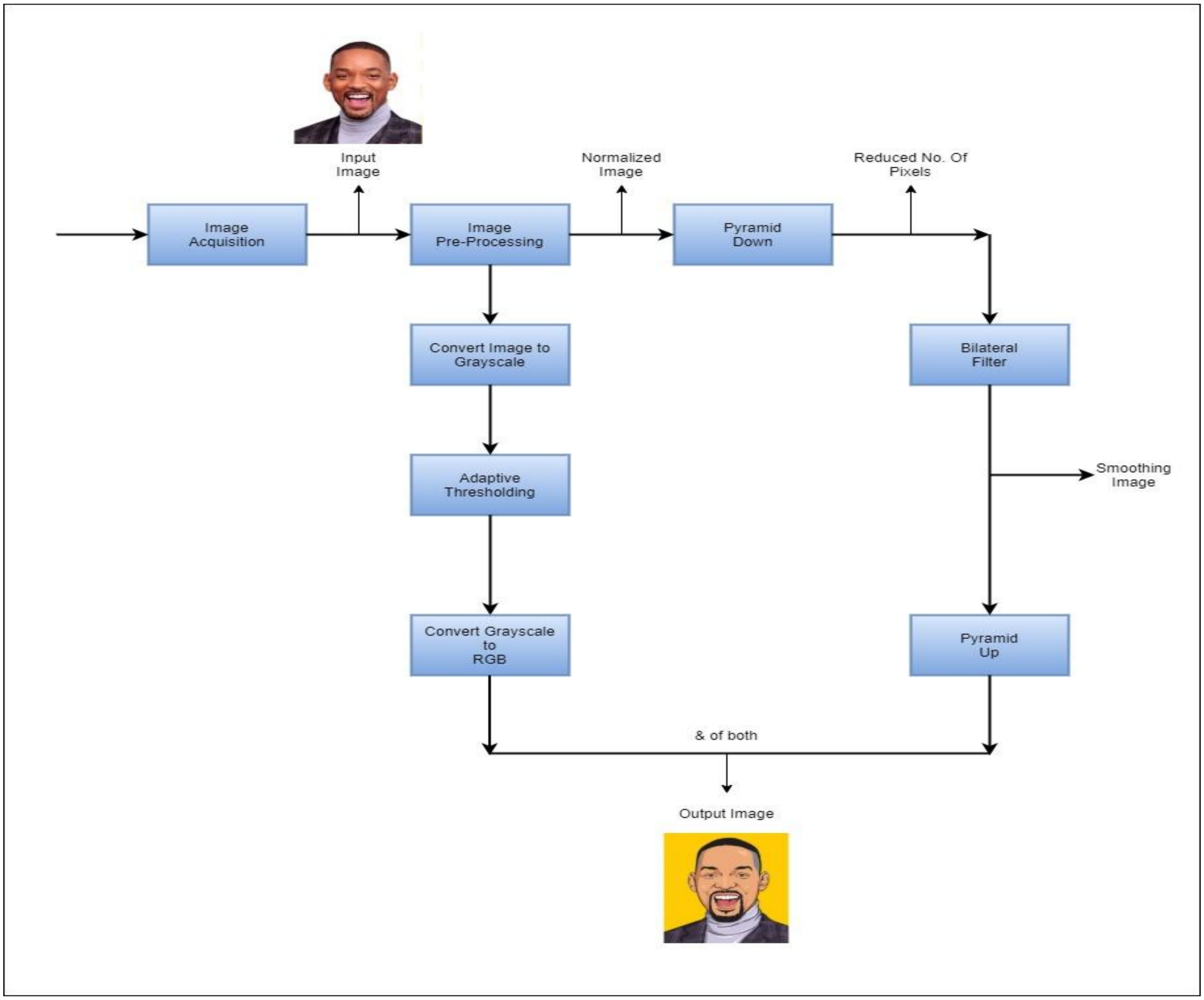
Dr. M Prabu

Group - 6

Apurva Rathore (M190376CA)
Nidhi Redekar (M190366CA)
Naziya Khanam (M190393CA)
Shivangi Kesharwani (M190402CA)
Saif Ali Khan (M190379CA)
Kundan Singh Bhadoriya(M190661CA)
Saurabh Shahi (M190378CA)
Anjali Sharma (M190374CA)

1.Block Diagram and Explanation:

Block Diagram :



Explanation:

To get the cartoonized image generally there are two main operations that we have to apply to get the output. One is to reduce the colour palette and the other is to identify the edges. Basically reducing the colour palette is done by the Bilateral filter, it refers to reducing the noise from the image and making the image smooth. To Identify edges we have to apply Adaptive thresholding. It's a process of converting an image into one particular pixel either white or black given a particular threshold. Combination of both the operations result in the image into cartoon effect.

2.Dataset collection and working procedure :-

In our project 'Cartooning of an image', we are converting a person's image to its cartoon aspect. We have taken the image dataset from https://www.kaggle.com/prasunroy/natural-images. This dataset originally includes 6899 images from which we have taken approx 400 images with size 16kb to 20kb and a dimension of 256 x 256 (width = 256 pixels and height = 256 pixels). The resolution of these images is 96 dpi and the bit depth is 24.

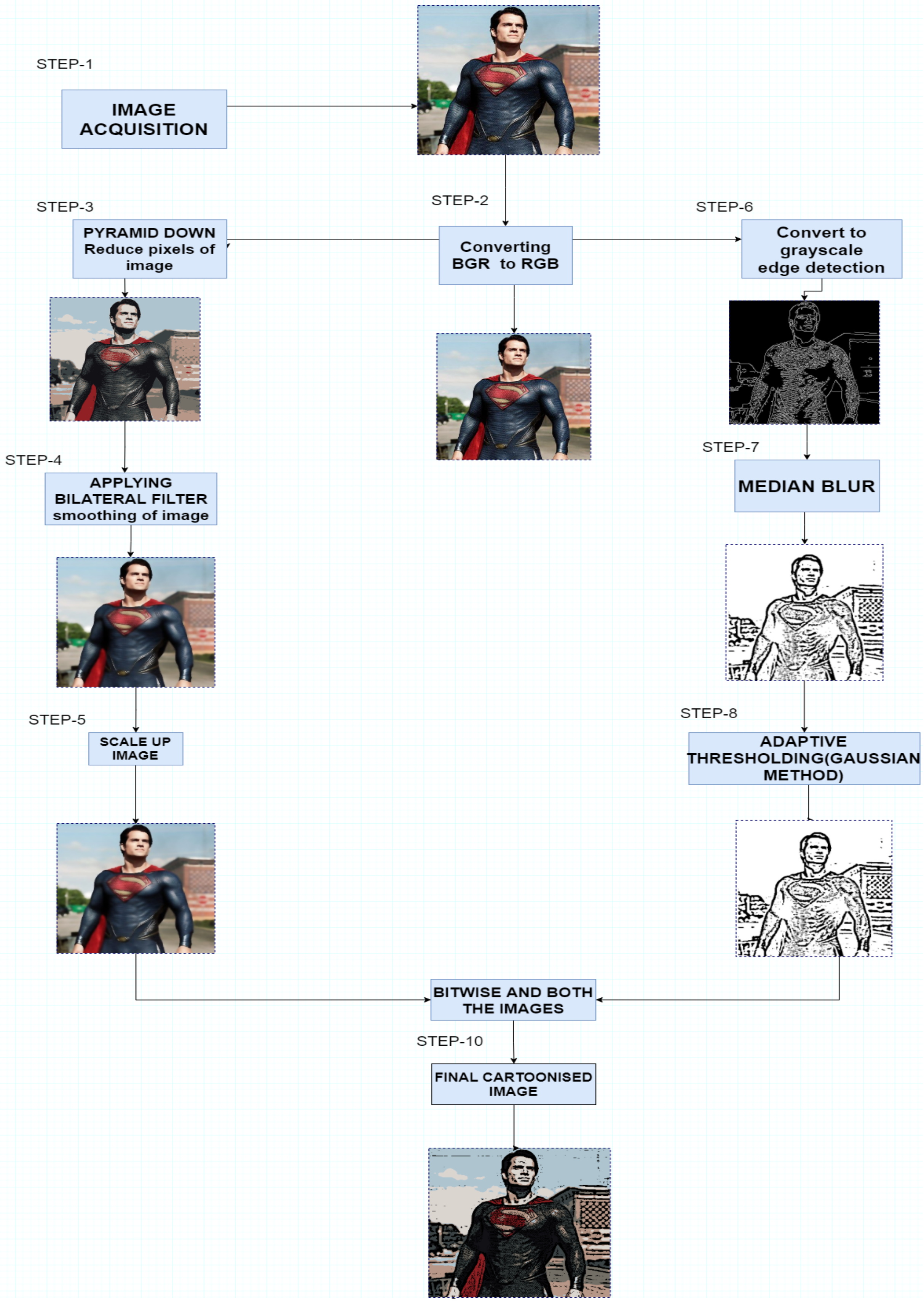
By using OpenCV in Python we are going to process our images and first, we will import all the essential libraries. Then we can read the images into a variable by providing the path of the images in our system using "cv2.imread(imagePath)." We will also print the images to verify that the images are correctly stored in a variable.

Now, as the image dataset is imported into the program, we can perform various operations for cartoonizing those images by reducing the color palette and edge identification through Bilateral Filter and Adaptive Thresholding and get the necessary output.

3. Methodology - (IP Processing and Methodology):

Diagram: A Block diagram of the Methodology that we have used is represented here in steps.

CARTOONIFY IMAGE


STEP-1
IMAGE ACQUISITION
STEP-3
PYRAMID DOWN Reduce pixels of image
STEP-2
Converting BGR to RGB
STEP-6
Convert to grayscale edge detection
STEP-4
APPLYING BILATERAL FILTER smoothing of image
STEP-7
MEDIAN BLUR
STEP-5
SCALE UP IMAGE
STEP-8
ADAPTIVE THRESHOLDING(GAUSSIAN METHOD)
BITWISE AND BOTH THE IMAGES
STEP-10
FINAL CARTOONISED IMAGE

Explanation:

Image Acquisition is the process of retrieving the image from the source. Images can be of any format which means in jpeg, jpg, or png format. OpenCV image reader function (imread()) reads the image as a NumPy array ndarray of row (height) x column (width) x color (3) and here the order of color is in BGR format, so image should be converted in the RGB format. After getting the image in a correct format,two operations are performed on this Image.

1. Image restoration is the operation on corrupt images such as noisy, camera mis-focus, blur images etc. and estimates the clean and original image.For Image Smoothing we have used the Bilateral filter which actually removes the high frequency content(noise,edges) and reduces the color palatte.

2. Detect the Edges Adaptive thresholding is used, and before applying this Median blur is performed so that it will help in later processing.Adaptive thresholding takes a grayscale or color image as input so RGB image has to be converted into Grayscale.For each pixel in the image threshold is calculated. OpenCV provides **adaptiveThreshold()**, it uses two methods, one is to find out the mean of neighboring pixels and other is to find out the total weight of neighboring pixels,where weight is the Gaussian window. At last we can combine the results of the first and second operation and we get the cartoon effect.

4. Classification : IP Algorithms with clear Explanation

1. Median Blur
2. Adaptive Thresholding

Median Blur : This is used to remove noise from the image, if in case the image contains any noise so it will restore that,so that in later processing a quality of image can be used.
In this, a median of each patch is found and replaces it with its mid Value. For example

234	67	245	234
123	45	23	245
34	0	234	24
255	235	145	253

In this image take a patch of 3*3 and find the median of it.
Values = 234, 67, 245, 123, 45 , 23, 34, 0, 234
Sort this value to find the median.
Values = 0, 23, 34, 45, 67, 123, 234, 234, 245

Median = 67
Do this for the entire image.

Adaptive Thresholding : It is a method in which a threshold is calculated and operation will be performed according to that. It is basically of 3 types:

1. Global Thresholding : In this a Threshold is fixed for entire image and operation will be performed according to that, for example :

   Threshold = 150, so all the values Greater than 150 assigned 255 or all the values less than 150 assigned 0.

2. Mean Threshold : in this a threshold is calculated for each pixel, by taking mean of its neighbouring pixels.for example :

234	67	245	234
123	45	23	245
34	0	234	24
255	235	145	253

This is an image, take a patch of 3*3 and find the mean of it and that is the threshold value and do this for each patch.
For 1st patch = (234 + 67 + 245 + 123 + 45 + 23 + 34 + 234 + 0)/9
Threshold value = 111.667

3. Gaussian threshold : The threshold value is a gaussian-weighted sum of the neighbourhood values minus the constant **C**.