NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

"CARTOONISE AN IMAGE USING BILATERAL FILTER"

A MINI PROJECT REPORT

Submitted by

RAMYA PRIYA Y (1NH18EC095)

TANMAYA S H (1NH18EC112)

NEHNA MANOJ (1NH18EC139)

NIKITHA M S (1NH18EC079)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

NEW HORIZON COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "**CARTOONISE AN IMAGE USING BILATERAL FILTER**" carried out by **NIKITHA M S(1NH18EC079), NEHNA MANOJ(1NH18EC139), RAMYA PRIYA Y(1NH18EC095), TANMAYA S H(1NH18EC112),** bonafide students of Electronics and Communication Department, New Horizon College of Engineering, Bangalore.

The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project Guide	HOD ECE
Ms. Monika Gupta	Dr. Sanjeev Sharma
Senior Assistant Professor	B.Tech, M.Tech, Ph.D
Department of ECE	
----------	----------

External Viva

Name of Examiner	Signature with Date
1.	
2.	

ABSTRACT

In this project we convert the image to cartoon picture. Which is one of the most used application in many industries. Cartooning is one in all foremost extensively used in gaming world animation or show biz. Converting the normal image to cartoon involves mainly three steps like detecting edges, smoothening the colors in the images and combining the edges on color images back to get the output. The images are converted into 2D type or like a print which gives that cartoon look. Many methods are used to detect the edges but canny edge detection method gives the better output. Bilateral filter (denoising filter) has many applications, smoothening of image is one of them. This filter gives the better output then the other filters. Applying the image repeatedly over bilateral loop again and again gives a better smoothened output of color image. Median filter is used to remove the noise present in the input images. Overlaying the edges on bilateral image is done using a Matlab keyword. This project can be applied to many real time images and get the output. Cartoon images are used in trending apps like instagram, video games and used in making movie. A code is developed in Matlab software for this application. Matlab tool is very useful to develop codes and see the output.

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **Ms. Monika Gupta** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

RAMYA PRIYA Y (1NH18EC095)

TANMAYA S H (1NH18EC112)

NEHNA MANOJ (1NH18EC139)

NIKITHA M S(1NH18EC079)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 01

INTRODUCTION

Cartooning a digital image is a very interesting and easy project to work on. The purpose of cartoonising an image is to convert a real image to a cartoon image so that it looks handwritten. It is a type of non-photorealistic rendering technique used in movies and video games for experimental animation and architectural illustration. It converts a 2-dimensional real image into a flat image to make it appear like it was drawn by a pen or brush to give artistic rendering of input image.

To achieve an animation picture from a digital image, the only need will be the bilateral filter and edge detection mechanism. These bilateral filters will assist us with reducing the shading palette or color of the image, which is an important step to obtain the animation look. Edge detection is used to gets perfect bold outlines. To give final cartoon effect the result obtained after applying bilateral filter and edge detection should be combined.

In this project, the image cartooning is obtained by application of bilateral filtering as edge preserving method. Cartooning is one in all foremost extensively used procedures, be it a gaming world, animation or the show biz. The main application of this effect is that it's widely utilized in video games and films where certain objects are made to appear like cartoons. In apps like Instagram and certain photo editing apps like pic toon where there is a wide usage of simple and fun photo effects this plays a vital role.

CHAPTER 02

LITERATURE SURVEY

Paper No	Title of the paper	Author & Year of Publication	Outcome	Limitation
1.	Bilateral filtering for gray and colour images	C. Tomasi and R.Manduchi 1998	It explains overview of bilateral filter.	It Doesn't give much information regarding smoothening of the image.
2.	Edge preserving smoothing	M.Nagao and T.Matsuyama 1979	Explains edge preserving.	It Doesn't give a clear picture on edge preserving.
3.	A Computational approach to edge detection	J Canny 1986	Helps us to understand canny edge detection technique	It doesn't give information about edge detection operators.

4.	Bilateral filtering as a tool for image smoothing with edge preserving properties	S. Srisuk 2014	Explains image smoothing	It doesn't give information about its operators.
5.	A study on image merge method using genetic algorithm for image based e-learning system	Y. Naganawa, H. Kawanaka and Tsuruoka 2008	Gives description about merging technique	Doesn't provide details about how to implement it.

Table 2.1

CHAPTER 03

EXISTING SYSTEM AND PROBLEM STATEMENT

Pic toon is a cartoon app which generates the cartoon face from the input image.

It consists of 3 major components, they are:

1. An image-based cartoon generator: It can generate a cartoon sketch from a given frontal face image with a little user interaction.

2. An interactive cartoon editor: Is an interactive tool through which the user to edit the expressions of a cartoon face.

3. Speech driven cartoon animator: It can automatically animate the cartoon face by just giving speech.

3.1 Problem Statement:

Real images can cause problems as edge detection is not very accurate with real images. A filter is used to obtain cartoonised image.

CHAPTER 04

PROPOSED METHODOLOGY

Over the last few years, many professional cartooniser software have popped up all over the world and also there are various online platform to convert an image to cartoon but in order to achieve the basic cartoon effect, neither powerful rendering software nor even years of experience is needed. All that is needed is essentially some edge detection and a bilateral filter which is implemented in this project using Matlab.

The command edge is used to extract the edges from the image. There are the different edge detector operators supported by Matlab like Sobel, Prewit, Roberts, Log, Zerocross, Canny operators out of which canny operator provides a better edge detection compared to all algorithms. The following table gives comparison of different edge detection operators.

Table 1 Experimental Images

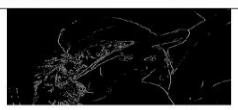
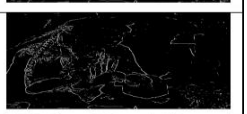
	Lena	Cameraman	Living room	Pirate
Original Image				
Canny				
LoG				
Prewitt				
Robert's				
Sobel				

Table 4.1: Comparison of different edge detection operators

Canny detection gives a best detection because the algorithm can mark as many real edges within the image as possible. It works fine under noisy condition. It provides a better localization by marking edges as close as possible to the edges within the real image. It provides minimal response i.e, a given edge in the image should be marked just once, and image noise must not create false edges.

4.1 Canny Edge Detection Algorithm:

It is observed that the "Canny Edge Detection Algorithm" is the only one optimal algorithm among all the edge detection algorithms in Matlab.

The three main steps in the "canny edge detection" are:

1. Low error rate: It's important that edge occurring in image must not be missed. There should be no response for non-edge also.

2. Good Localization: The space between the edge pixels as found by the detector and the real edge should be minimum.

3. Single Response: It should have one response to one edge. The canny algorithm mainly has five steps in it.

They are as follows:

Step 1: Computing the vertical (Gy) and horizontal (Gx) gradient of every pixel in a picture.

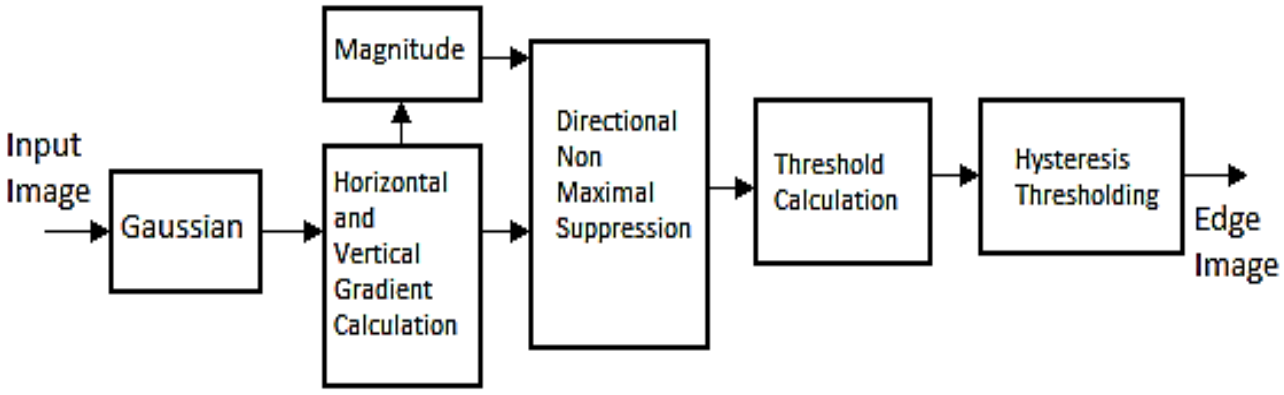
Step 2: By using the above information the magnitude (G) and direction of each pixel within the image is calculated.

Step 3: In this step all non-maxima's are made as zero that's suppression of the non-maxima's hence this step is known as "Non-Maximal Suppression".

Step 4: The low and high thresholds are measured by using the histogram of the gradient magnitude of the image.

Step5: To get the correct edge map hysteresis thresholding is used which can link the weaker and stronger edges. The weak edges should be taken into consideration if and only if it's connected to the strong edges otherwise else it should be eliminated from the edge map. The edge is claimed to be strong edge if its pixel value is greater than the high threshold else it's said to be weak edge whose pixel value lays between high and low threshold.

Then by using that, the pixel strength and orientation of that gradient is computed. Within the next step it finds the all maxima's present in the image then it keeps them as it is, and then removes the remaining non-maxima's which don't seem to be necessary. Within the step 4 it makes the pixel either the non- edge or edge, based on the low and high thresholds set. The block diagram of the "canny edge detector" can be seen in the figure 1(a).



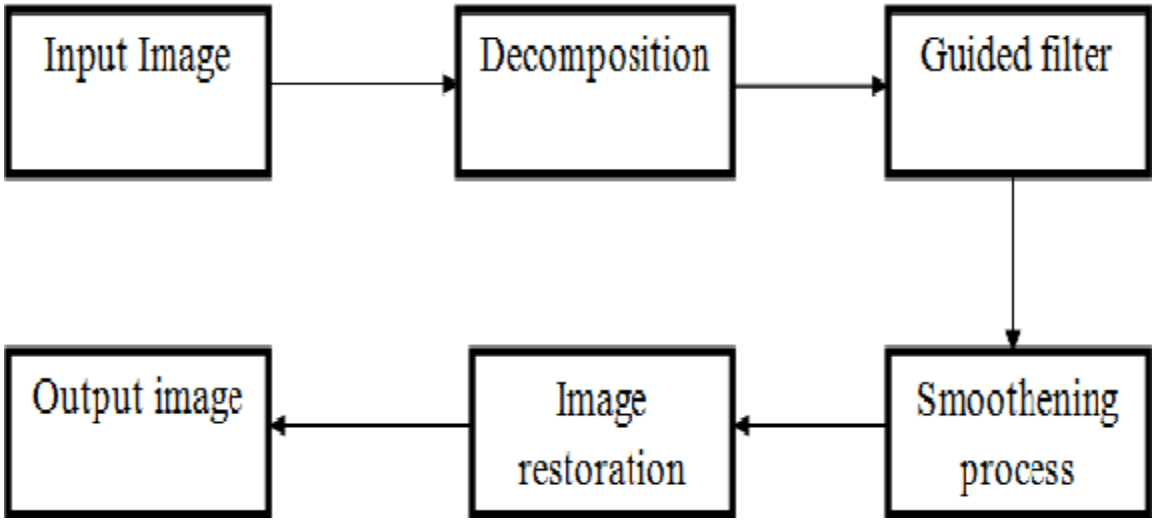
Chap 4. Fig 1(a): Block diagram of canny edge detector

If the pixel has higher value than the high threshold then it's treated as edge. If pixel has value less than the low threshold then it's treated as non-edge. If pixel value is in between the high and low thresholds then it's a weak edge. Thus, to detect edges within the image it considers two thresholds as high and low.

Then the hysteresis thresholding is applied which can help to make a decision that whether the detected weak edge should be considered or not. The pixel is compared with the neighboring pixels and if the weak edge is connected to strong edge then it's considered as edge or else it will be removed from the edge map. The threshold value is same for all the photographs. Hence it has some limitations, when applied to the block level of the image. It provides some false edges within the plain region and sometimes it fails to detect some significant edges. So as to avoid the above limitation an adaptive thresholding block and also the block classification blocks are also added together with the above blocks and also the threshold is made different based on the block. Therefore, the performance of the proposed block level canny edge detector is improved.

4.2 Bilateral filter:

Bilateral filter is also known as the key element in the colour image processing chain, because it homogenizes colour regions, even over multiple iterations. It is an edge-preserving filter which has very similar response to the Gaussian filter called as bilateral filter. The property of smoothing image while preserving the edges of the image makes the it a really popular image processing tool.



Chap 4. Fig 1(b): Block diagram of bilateral filter

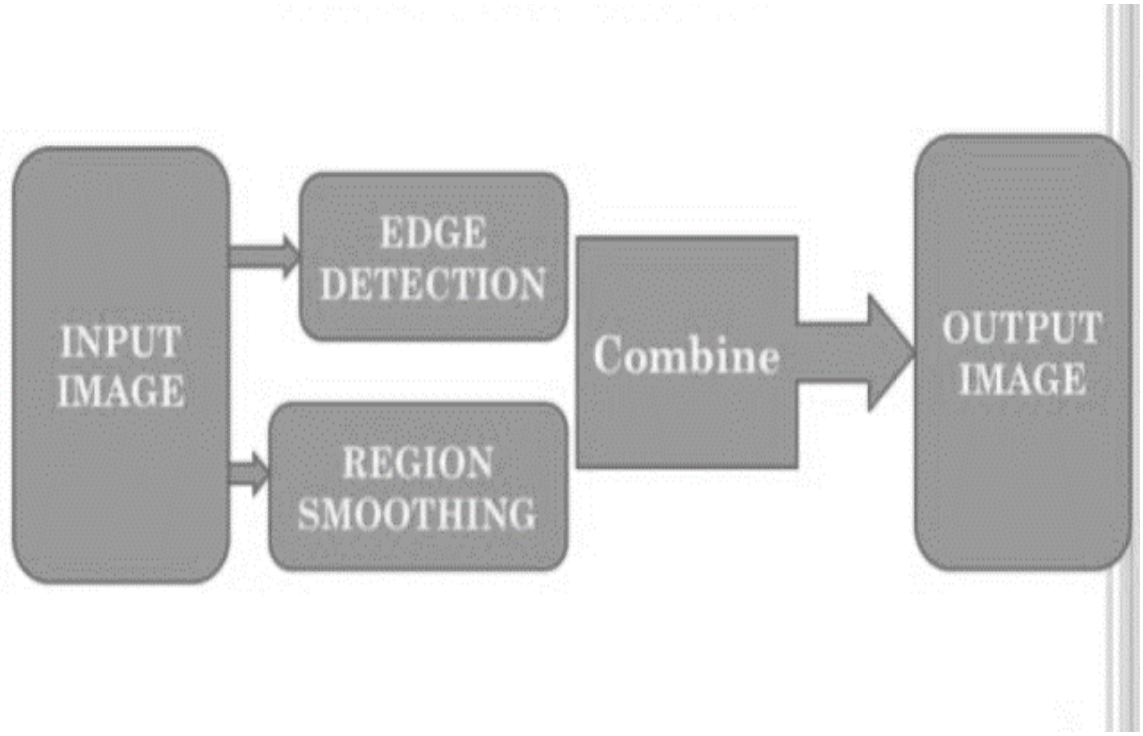
It smoothens the image while preserving edges, by using non-linear combination of nearby image values. It merges gray levels or colors depending on both their photometric similarity and their geometric closeness therefore prefers near values to distant values in both domain and range. Hence within the output image each pixel is replaced by a weighted average of its neighbors and averaging is carried out in range and spatial technique. So, the output obtained by this is the smoothened image whose noise is also reduced.

The edge detected image is overlayed on colour image output of bilateral filter using imfuse Matlab keyword to get the specified cartoonised image.

To create blended overlay image and to scale the intensities of q and w jointly as a single data set, the following code is used

y=imfuse(q,w,'blend','scaling','joint');

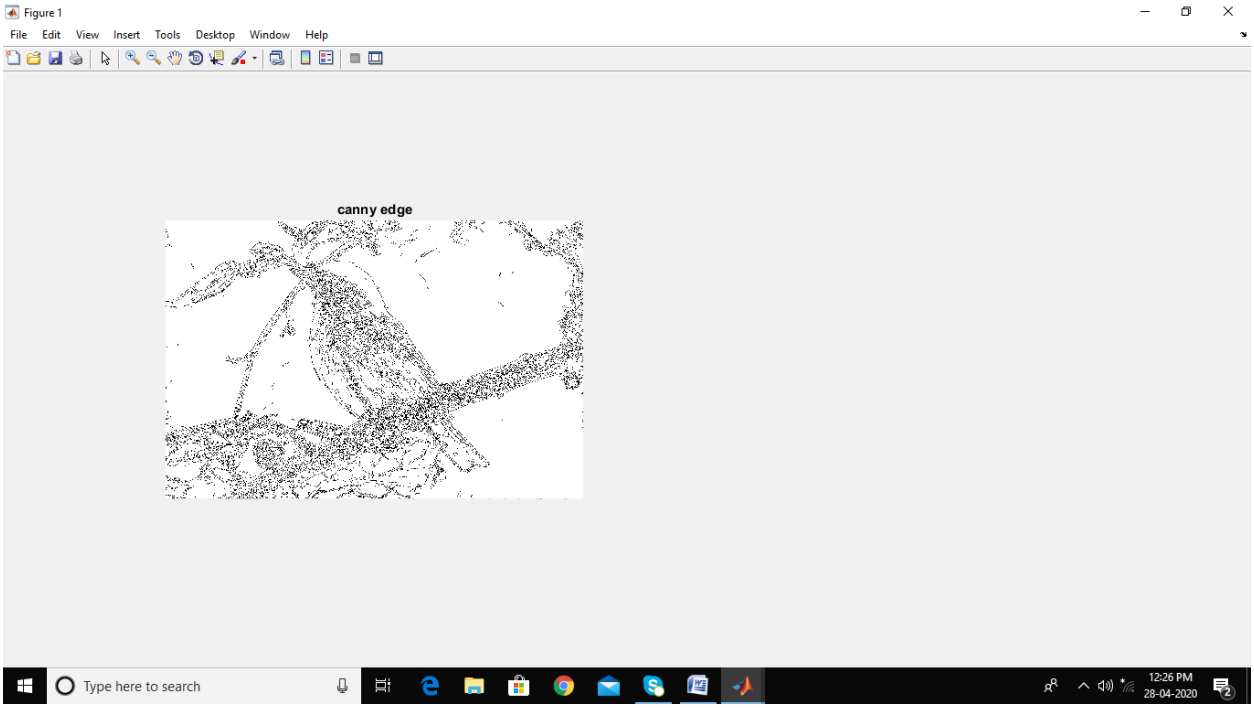
To view the fused image imshow(y) is used.



Chap 4. Fig 1(c): Block diagram of cartoonising an image

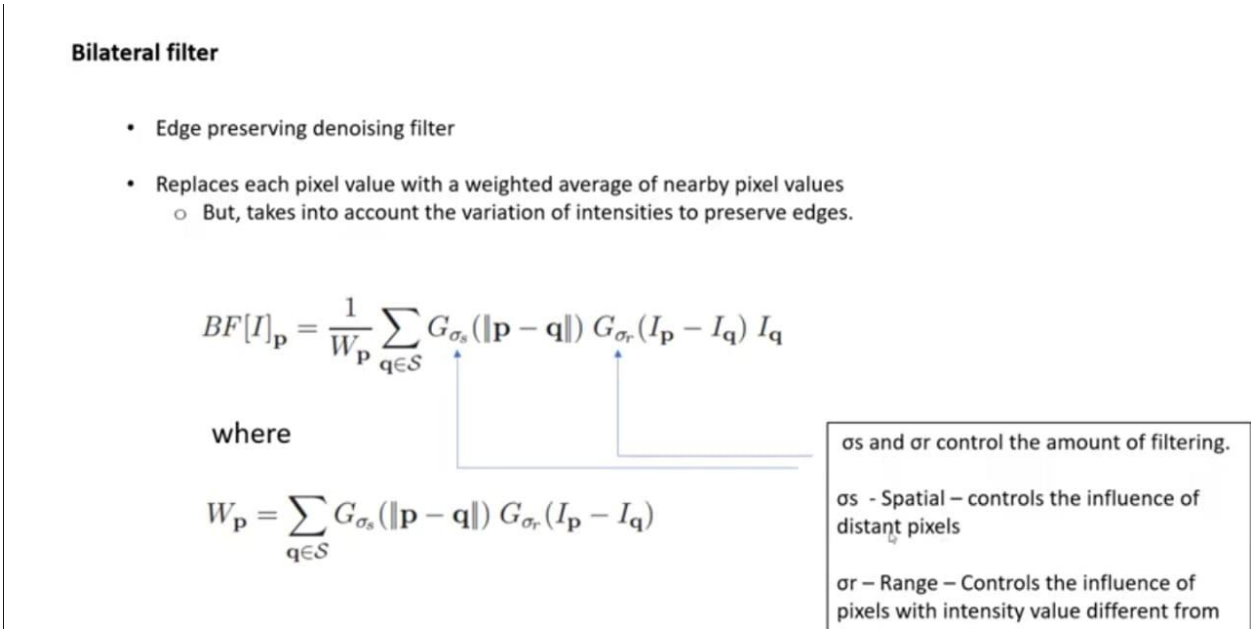
4.3 PROJECT DESCRIPTION

In this project we convert a normal image into cartoon image. To get a cartoon image, an original image has to undergo blocks such as edge detection, bilateral filtering and overlaying of images. In edge detection, the edges are where the intensity changes are detected and marked. A median filter is applied before edge detection to remove noises in the image for best output. Canny edge detection is one of the best ways to detect the edges of color image. Edge is detected by applying the derivative in which ever axis it is present. First derivative finds the maximum of edge and second derivative finds the zero of the edge. Derivatives are found by applying filters on horizontal and vertical directions. In canny edge detection, the Gaussian filter is applied to find the edges with standard deviation sigma. Then the computation of the gradient in x-axis and y-axis is done to get magnitude and direction. A lot of edges will be detected, so we apply NMS (non-maximal subtraction) to make edges to look thin and then hysteresis thresholding is applied, so that edge with a more intensity than the threshold value is detected.



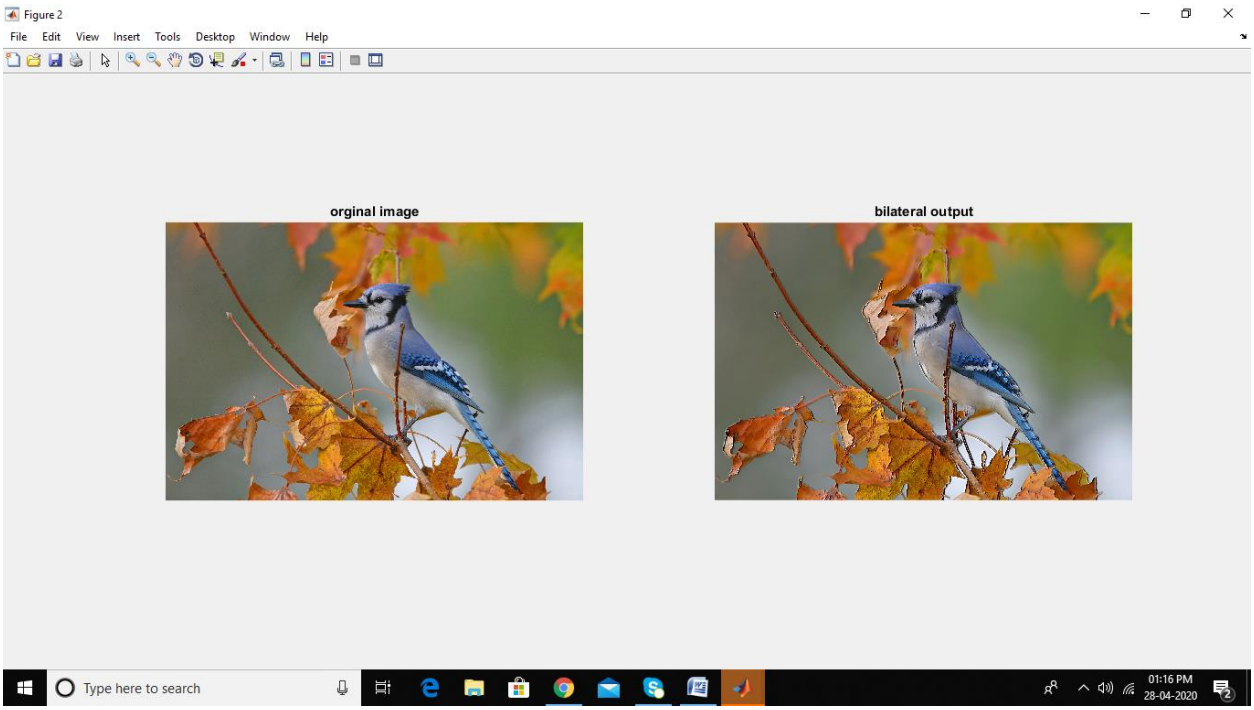
Chap 4. Fig 2(a): Canny image detection

Bilateral filtering is used to smoothen the colours of image . It is a denosing filters which is used for most of the applications . It places each pixel value with a weighted average of nearby pixel values but also takes into account the variation of intensities to preserve the edges. Develop the code for bilateral filtering based on the formula.

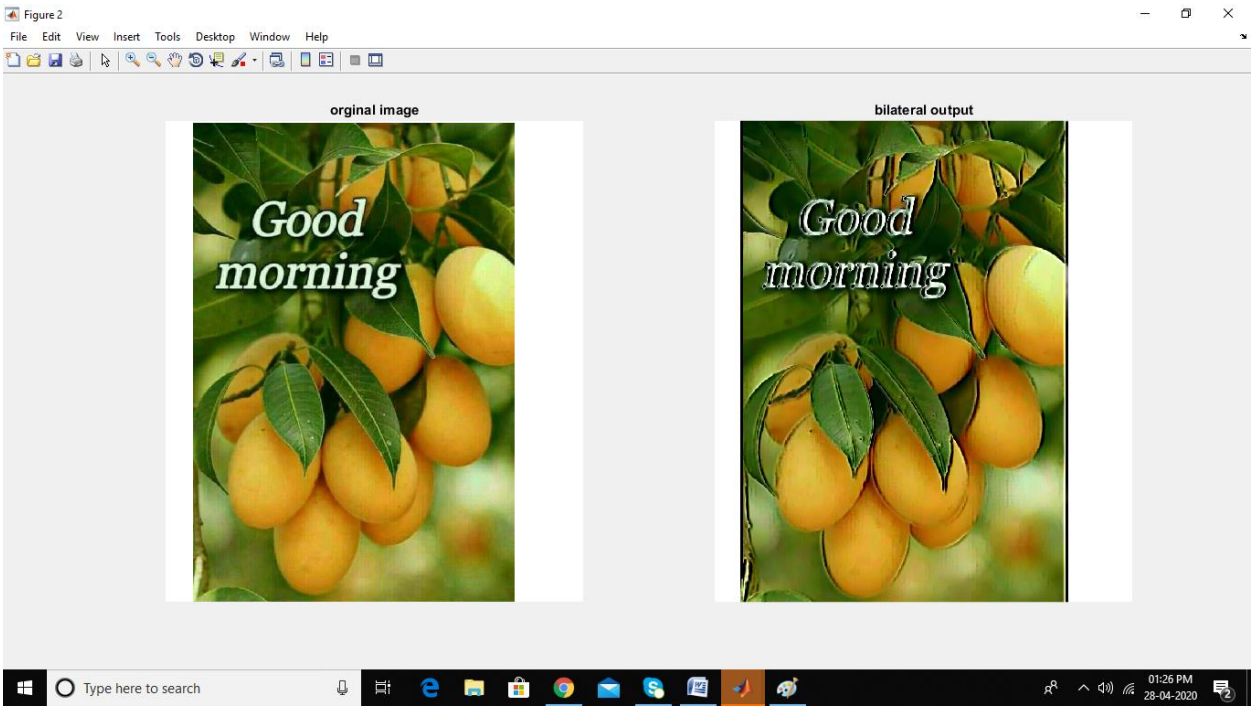


Chap 4. Fig 2(b): Bilateral filter formula

Wp is the weighting average of nearby pixel,sigma s and sigma R controls the amount of filtering. Bilateral fiter is applied in a form of a loop . The variations value , mask value and no of repeations are mentioned in the code. If we repeat the smoothening of images many times it gives the better output. Convert the RGB image to ycbcr (it contents main information of the image) . Then bilateral loop to this ycbcr image type , later convert it back to rgb form.



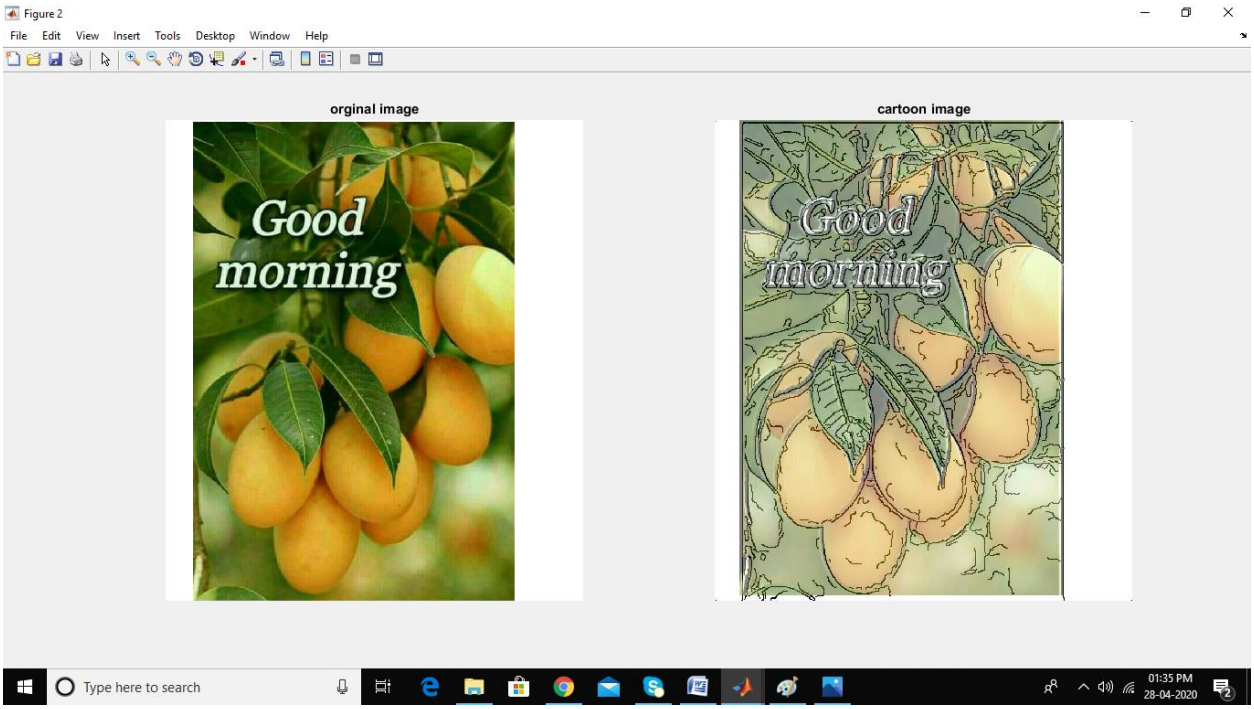
Chap 4. Fig 2(c): Bilateral filter output



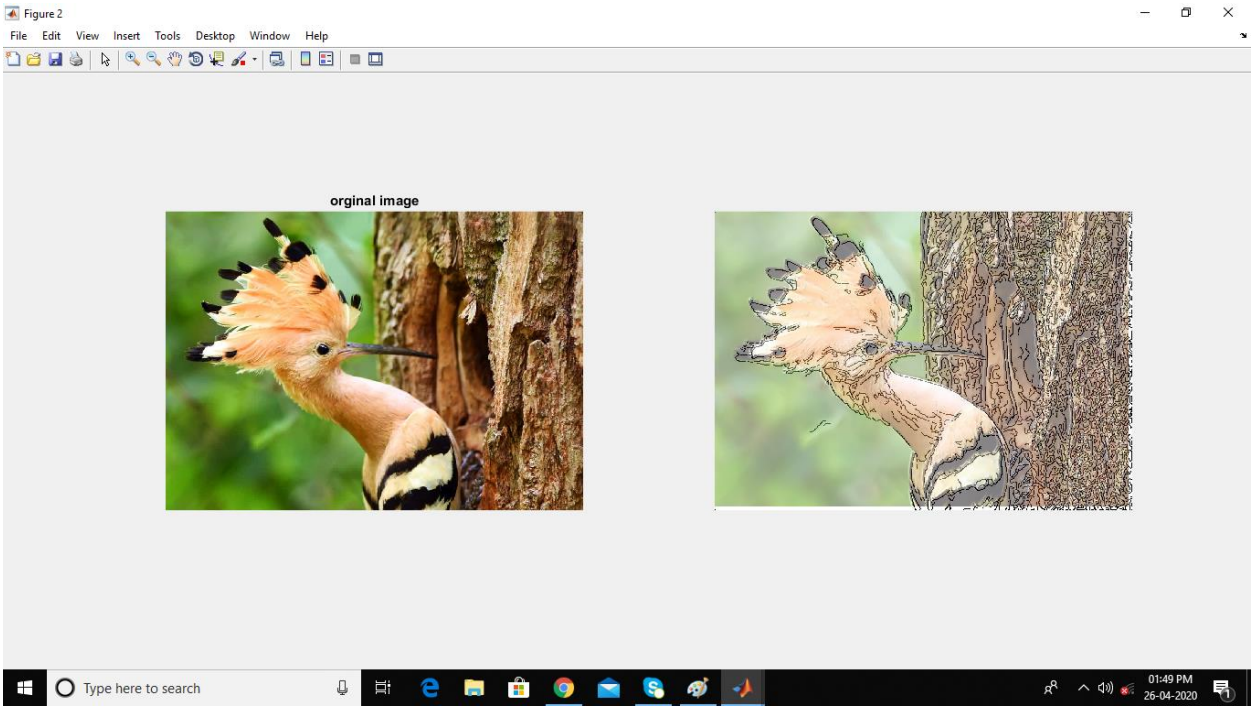
Chap 4. Fig 2(d): Bilateral filter output

Overlaying of images is done by using matlab keyword imfuse.Syntax:

y=imfuse(q,w,'blend','scaling','joint');



Chap 4. Fig 2(e): Cartoonised image



Chap 4. Fig 2(f): Cartoonised image

CHAPTER 05

SOFTWARE DESCRIPTION

1. MATLAB 16.0:

Matlab 16.0 is the software tool used for writing code for this application. It is developed by the MathWorks. Matlab stands for matrix laboratory. It is designed by Cleve Barry Moler. It allows matrix manipulations, implement algorithms, plot of data. It takes an image as matrix with rows and columns. Mathematical operations are applied on the images. It also deals with programs written in other languages.

Advantages:

- Test and execute of the program written is easy.
- Debugging is easy. It has built in algorithms.
- Processing of images and simulating of videos is easy.

Disadvantages:

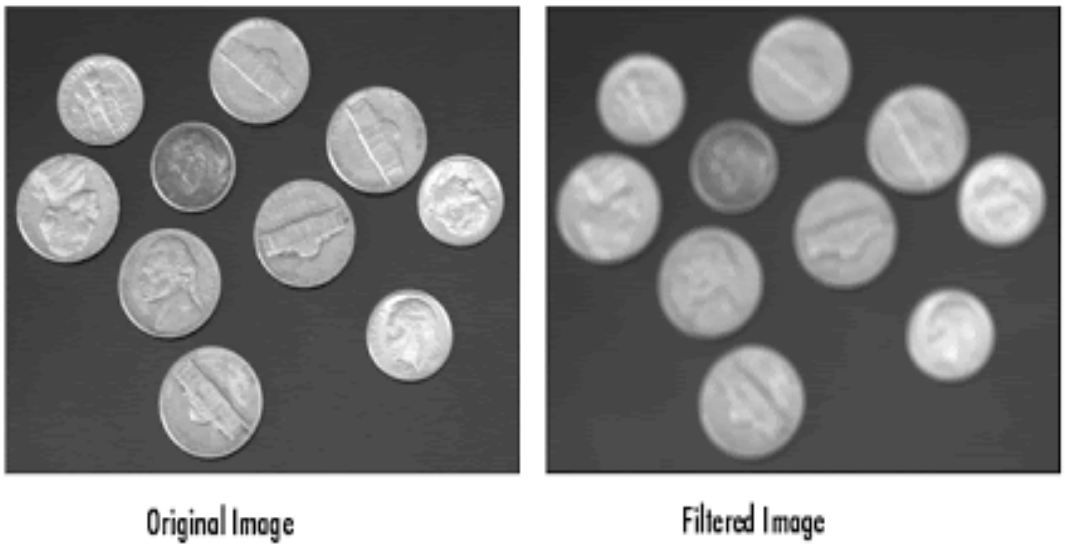
- Matlab compiler and packaging tools for add on are not supported.
- Xlsread and Xlswrite will work only in basic mode
- The cost of original license for matlab is very high.

Some of the functions used in the code:

- A = imread (filename, FMT) Specifies the format of the file that contains the common file extension specified by FTM. If Imred cannot find the file with the filename, it will look for the file name filename. Fmt.
- imwrite:
  - The file name A writes to the file, including the file format of the Image imwrite (A, file name) extension. The output image depth depends on the data type A and the file format.
  - imwrite (A, map, filename) writes the index image to the map on the file specified by A and its associated colormap file name.
- imshow(a) is used to display the image.
- C = imfuse (A, B) creates a composite image from two images, if A and B are of different sizes, then small-sized images with Zeros create impulse pads, so that the two images must be the same before fusing. The output, C, is a numerical matrix in which the combination of A and B images occurs.
- X = zeros (sz1, ..., szN) sz1-by -..- szN Returns the range of zeros, where sz1, ..., szN represents the size of each dimension.
- X = size (A) gives a series vector whose elements are the lengths of the corresponding dimensions of A.
- SE = Stellar ('Rectangle', [M N]) Forming an element with a rectangular structure of size.
- X = imperfection (I) computes the complement of the image I and gives the results in X.

- X = imdilate (I, SE) dilutes a grayscale, binary, or packed binary image I, returns a thin image, X. SE is a structured element object or array of structured element objects, returned by std or offsetstrel functions.
- X = imedilate (I, SE) is a grayscale, binary thinning or binary image, and returns a thin image, X. SE is a list of build objects or configuration objects, which is a strel or offsetstrel function.
- X = Seal (Y) round each element of Y to the nearest integer greater than or equal to that element.

2. FILTERING:



Chap 5. Fig 3(a): Filtered image

When an image is acquired by a camera, it is usually a visual system which it is designed for and it cannot be used directly. The image can be randomly damaged size differences,

light differences or negative differences to be dealt with in the first stages of vision correction. Filtering is probably the most important base operation

processing and computer vision. In the broad sense of the word filtering the value of

a filtered image in the area provided by the function of a small input image in a small neighborhood of the same place. As mentioned earlier, images are often corrupted by random variations in the values of the intensity called noise. Some common types of noise are salt noise and pepper noise, impulse noise and Gaussian noise. In contrast, the Gaussian noise difference in magnitude drawn in a Gaussian or normal distribution is very good model for many types of sensor noise, such as noise due to electronic cameras. Linear filter is a good filter for removing Gaussian noise and in many cases, other types of noise again. The use of a linear filter is applied to a weighted sum of pixels respectively windows. Usually, the same weight pattern is used in each window, that is the direct filter is localized and can be performed using the convolution mask. If different filter elements are used for different parts of the image, but the filter still functions as a dynamic load, then the linear filter varies. Any filter that is not a weighed sum of Pixels is a non-linear filter. Nonlinear filters can invade each other over time, meaning the same the calculation is done without any position on the image or the spatial variations.

3. GAUSSIAN FILTER:

Chap 5. Fig 3(b): Gaussian filtered image

Gaussian filtering is a category of smooth nonlinear filtering with components selected according to the Gaussian function formation. A smooth Gaussian filter is an excellent filter to remove noise distribution. Gaussian nonlinear function in one-dimension pass filters by looking at both domains and quantities, they work well and can be used successfully by engineers in the practical visual applications.

Large Gaussian filtering can be performed very well because Gaussian functions exist inseparable. A two-dimensional Gaussian convolution can be performed by convolving the image with one dimensional Gaussian and then convolving the result with the same one-dimensional filter oriented orthogonal to the Gaussian used in the first stage. Therefore, the value of computation required for 2D Gaussian filtering expands directly to the filter range instead of growing in sequence.

Some of the Advantages and Limitations of Gaussian Filter

Advantages:

- It reduces noise.
- It helps in reducing contrast.
- It also blurs the edges.

Limitations:

- It is a slow process.
- It cannot preserve edges.

In particular Gaussian low pass filtering they include the average pixel values in the neighborhood, where weights decrease with distance from the neighboring center.

Audio values that damage these nearby pixels are less connected than the signal.

4. BILATERAL FILTER:

Chap 5. Fig 3(c): Bilateral filtered image

This filter is a key element in color image processing, as it enlarges regions of color in the process to preserves edges, over multiple iterations because it is an expensive work, the image is reduced by a factor of 4 in both x and y directions before they are filtered. The bilateral filter operates in the same way as a Gaussian filter since it gives each pixel a weighted pixel value in the neighborhood. However, the difference is that the weights are further refined depending on how different the pixel values are. This way, the pixel is closer to the color in a centroid pixel

and it will be heavier than a pixel in the same range in a different color. This is extra step in weight calculation is important because it means that sharp changes in color (edges) can be observed, in contrast with simple Gaussian blur. However, the duration of operation of a bilateral filter depends on the kernel size, and experiments have shown that using more iterations with smaller kernels generated that were more pleasing, and sooner rather than later is expressed using a bilateral filter located in the assembly with the largest kernel size for less iterations. The algorithm uses a 9 × 9 kernel and fills the filter 14 times in the image. When filtering is complete, the image is restored to its original size using linear interpolation to fill in missing pixels.

Some of the Advantages and Limitations are:

Advantages:

- It preserves edges.
- It is very simple and easy to use.

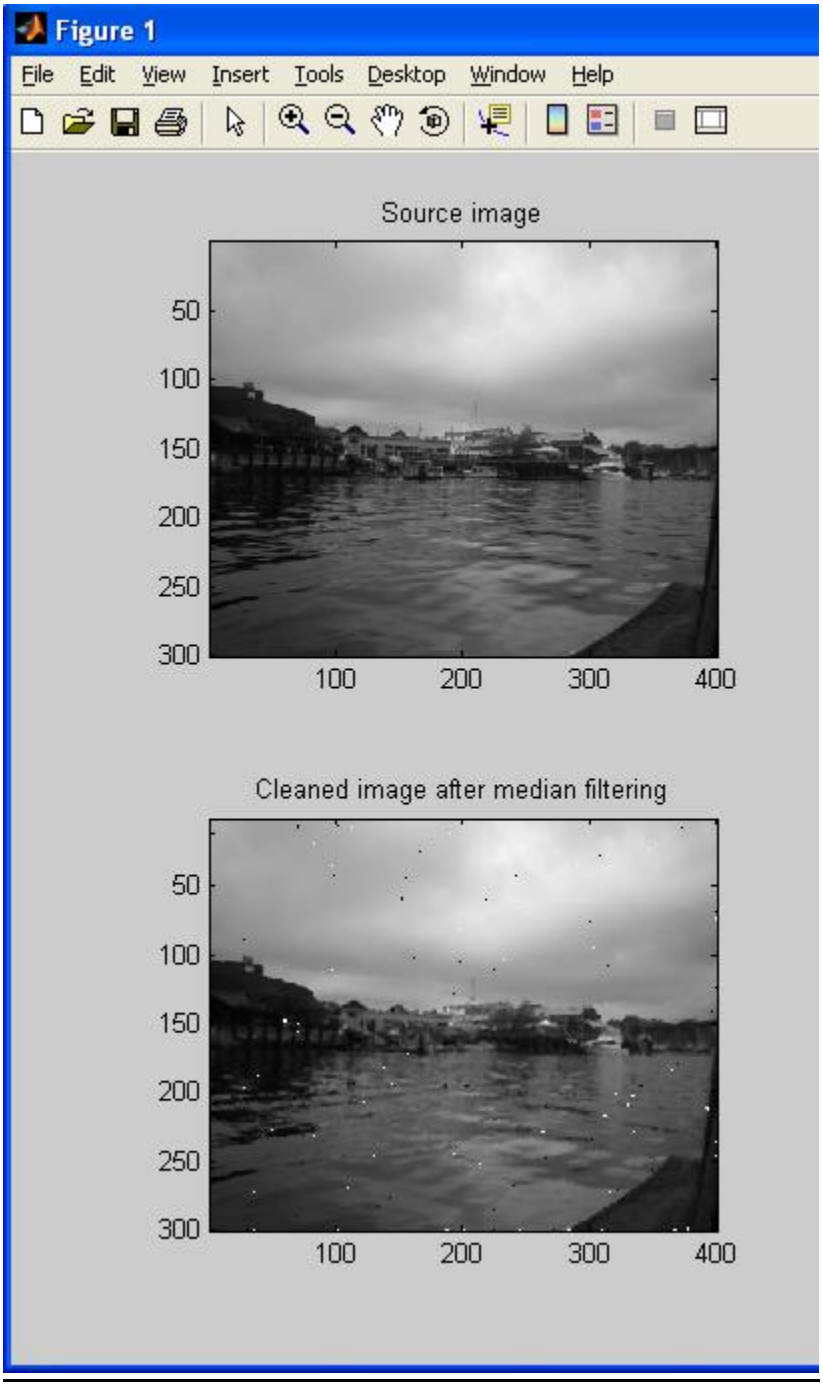
Limitations:

- It somewhere gives a false edge of the image.

APPLICATIONS:

- It is used in cartooning and tone management.

5. MEDIAN FILTER:



Chap 5. Fig 3(d): Median filtered image

Images are often affected by noise because they are subject to various distortions in digital, time search, processing, compression, storage, transmission and reproduction, which can produce noise. The basic task of image processing is to eliminate noise and restore the original image.

Therefore, the problem of getting the original image from the noise data has increased in recent years. When transferring an image, transmission problems cause the signal to run, giving a wrong value to one of the three points. This is the type of transmission defect known as "salt and pepper" noise from light and dark areas from image due to noise. The ratio of the number of incorrectly transmitted points is called the noise composition of the image.

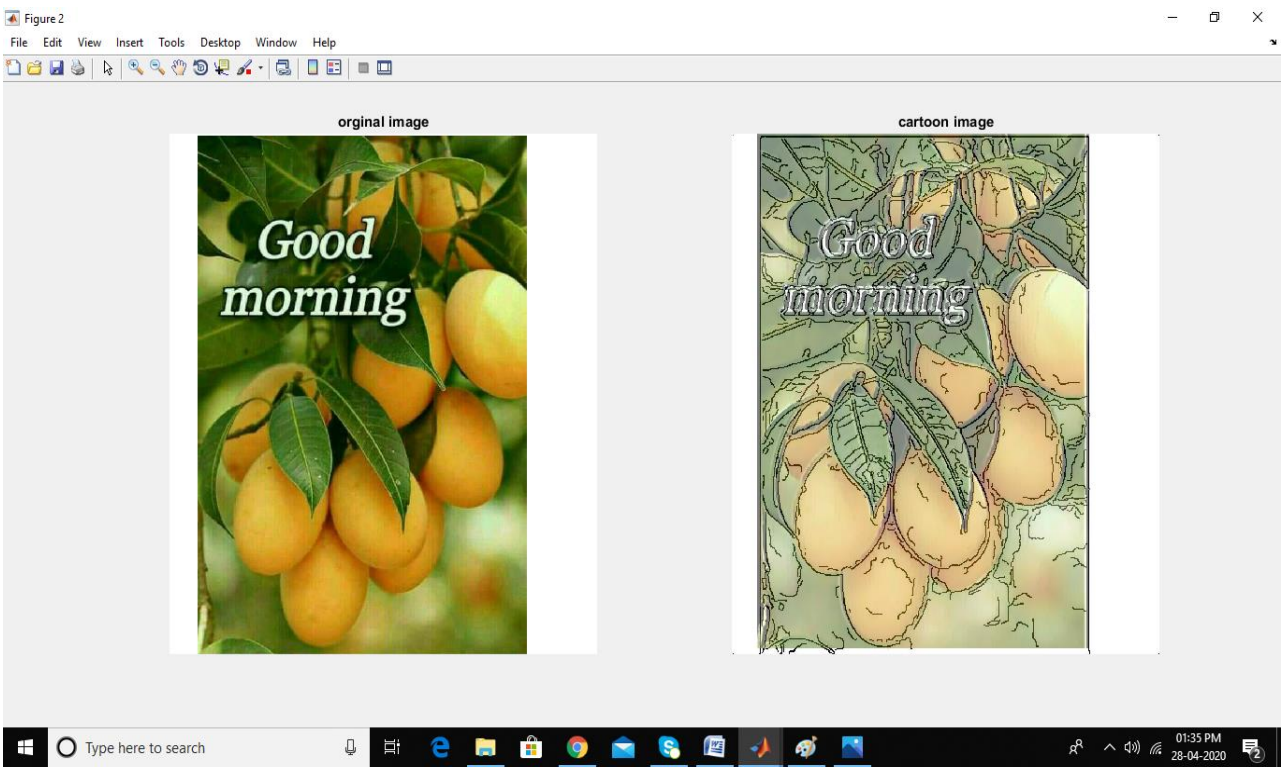
The goal of noise removal filters is to take the distorted image as input and generate roughly the output of the image without the knowledge of the noise or characteristics of the image's texture.

CHAPTER 06

RESULT AND DISCUSSION

6.1 RESULT:

Bilateral filter output is shown in the figure for a color image. Here in the output image each pixel is replaced by its weighted average of its neighbors and averaging is done both in terms of spatial and range. Hence the result image is smoothened and noise is minimized.



Chap 6. Fig 4(a): Cartoonised image (Output)

6.2 DISCUSSIONS:

- We propose a new way of creating a filter that keeps the edge which has a very similar response to a Gaussian filter known as bilateral filter.
- Bilateral Filter Optimizes image while preserving edges, using an uneven combination of adjacent image values. It includes gray levels or colors based on their geometric proximity and their uniformity of format preferring values that are closer to values that are away from the domain and range.
- The filter can use the visible metric under the color space as well the colors are smooth and retain the edges in a way that is permissible in human eyes.
- Also, unlike regular filters, bilateral filters do not make phantom colors along the edges of the color images and reduces the phantom colors thereby appears in the real picture.
- We have introduced cartooning which is an application of bilateral filter operation.
- Creating a cartoon is one of the most commonly used processes, be it gaming, animation or the entertainment industry.

We observed that,

- The output image each pixel is replaced by its weighted average of its neighbors.
- Averaging is done both in terms of spatial and range.
- The result image is smoothened and noise is minimized.

CHAPTER 07

ADVANTAGES AND APPLICATIONS

7.1 ADVANTAGES:

- More sharpness and better visual appearance for images.
- Minor errors can be corrected.
- You can adjust the image size.
- Images can be sorted automatically based on the content they contain.
- Images are sensitive.
- It allows industries to remove defective products from the product line.
- It enables weather forecasting.

7.2 APPLICATONS:

- It is used in the movie industry.
- It is used in video processing.
- It is used in machine or robot vision.
- It is used in video games for experimental animation and architectural illustration.

CHAPTER 08

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION:

- For the conceptual understanding of the present review, we introduce bilateral filters, their uses and their variants, and the advanced methods used to estimate it.
- We believe that the success of bilateral filters combines with its simple combination, good results and efficient algorithms.
- While there are options at every step, some, if any, combine all of these benefits.
- It is a highly efficient filter and is very flexible because it can change the weight of the list to fit any idea, including contrasting colors about pixel differences.

Limitations:

- Once the system is damaged the image will be lost.
- It is time consuming.

8.2 FUTURE SCOPE:

- Automotive Industry: In developing advanced driver less cars and they assist in autonomous vehicles and are widely used in automatic cars.
- Image enhancement: Camera apps on smartphones and digital cameras use image processing to enhance image quality, video stabilization and noise removal etc.

- Robotics: Navigation of in-flight robots in an unknown location (SLAM), control of a robot by processing video feeds from a camera to a robot to extract the live surroundings.
- Gameplay: Advanced game development such as the Xbox kinect uses image processing from human player movement analysis.
- Direct solutions to the problem: image processing is used as a solution to a variety of problems, from facial recognition to signal acquisition in the manufacturing industry.
- Productivity: Detecting errors in processes and controlling robots in performing specific tasks. Ex. Problems in creating a Printed Circuit Board (PCB) can be viewed using high resolution image processing.
- Human machine interface: machines are made intelligent by adding a gestural interface, or response space for a human gesture, which determines the action of a human being to perform certain tasks.

REFERENCES

[1] T. L. Huntsberger and M.F. Descalzi, “Color edge detection”, Pattern Recognit. Lett., vol. 3, no. 3, pp.205-209,1985.

[2] D. Marr and E. Hilderth, “Theory of edge detection”, Proc. Royal Soc. London, vol. B207, pp. 187-217, 1980.

[3] M. Nagao and T. Matsuyama, “Edge preserving smoothing”, Computer Graphics and Image processing, 9:394-407, 1979.

[4] C. Tomsai and R. Manduchi, “Bilateral Filtering for Gray and Color Images “, Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India.

[5] R. Gonzalez and R. E. Woods, “Digital Image Processing “,2nd edition, Prentice Hall,2002.

[6] B. Zhang and J. P. Allebach, “Adaptive bilateral filter for sharpness enhancement and noise removal”, IEEE Trans. Image Process., vol. 17, no. 5, pp. 664–678, May 2008.

[7]G. Ramponi, "A rational edge-preserving smoother", Proceedings of the International Conference on Image Processing, volume 1, pages 151-154, Washington, DC, 1995

[8]N. Himayat and S.A. Kassam, "Approximate performance analysis of edge preserving filters", IEEE Transactions on Signal Processing, 41(9):2764-77, 1993.

[9]Kornprobst, Pierre (2007). "Limitations? - A Gentle Introductionto Bilateral Filteringand its Applications" (PDF). Retrieved 7 May 2017.

APPENDIX

```
clc;
clear
close all;% edge detection
img=imread('ele.png');
[M N l]=size(img);
img_filtered= img;
for x=1:l
img_filtered(:,:,x)=medfilt2(img(:,:,x),[7,7]);
end
[M N l]=size(img_filtered)
dup=zeros(M,N);
for k=1:M
   for l=1:N
      dup(k,l)=img_filtered(k,l);
   end
end
fltr=edge(dup,'canny');
sel=strel('rectangle',[2 2]);
ca=imdilate(fltr,sel);
ca=imcomplement(ca);
```

```
%subplot(1,5,1);

%imshow(ca);

imwrite(ca,'gf.jpg');

title('canny edge');

%filter

a=imread('ele.png');

b= size(a);

b1=size(a);

a1= rgb2ycbcr(a);

a=a1(:,:,1);%taking a main component

a=double(a);

n=11;%inislaition

n1=ceil(n/2);

vars =50;

varr =25;

c=0;

c1=0;r=1;% filter loop

for i1=1:r

   for i=n1:b(1)-n1

   for j=n1:b(2)-n1

      for k=1:n

         for l=1:n
```

```
            c=c+gs(sqrt((-n1+k)^2+(-n1+l)^2),0,vars)*gs(a(i-n1+k,j-n1+1),a(i,j),varr)*a(i-n1+k,j-
n1+l);
            c1=c1+gs(sqrt((-n1+k)^2+(-n1+l)^2),0,vars)*gs(a(i-n1+k,j-n1+l),a(i,j),varr);
          end
        end
        d(i-n1+1,j-n1+1)=c/c1;
        c=0;
        c1=0;
    end
    end
a=d;
clear d
b=size(a);
end
d1=uint8(a);
d1(:,:,2)=a1(r*n1-(r-1):b1(1)-(r*n1),r*n1:b1(2)-r*n1+r-1,2);
d1(:,:,3)=a1(r*n1-(r-1):b1(1)-(r*n1),r*n1:b1(2)-r*n1+r-1,3);
figure;
subplot(1,2,1);
imshow(uint8(ycbcr2rgb(a1)));
imwrite(uint8(ycbcr2rgb(a1)),'sf.jpg');
title('orginal image');
```

```
imwrite(uint8(ycbcr2rgb(d1)),'bf.jpg');

j1=ycbcr2rgb(d1);

 q=imread('bf.jpg');

w=imread('gf.jpg');

y=imfuse(q,w,'blend','scaling','joint');

subplot(1,2,2);

imshow(y)

function out = gs (p ,q, var)

    % d=size(p);
     out = 1/((2*pi)*var^2)*exp(-(p-q)*(p-q)'/(2*var^2));
return
```