RV COLLEGE OF ENGINEERING®
BENGALURU – 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

MATHEMATICAL MODELING

EXPERIENTIAL LEARNING

Cartoonify Images using Mathematical Modeling: Detecting edges and applying various filters

Submitted by:

Anshul Jain **1RV19AS007**
Sudhanshu Garg **1RV19CS164**

To

Prof. Venugopal K
Assistant Professor
Department of Mathematics
RV College of Engineering

Objective

The goal of this project is to allow users to apply mathematical modeling to images of their choice. The model is designed to provide artistically and comically appealing results on as wide a range of pictures as possible, although it is conceded that not all inputs will yield equally satisfying results.

Methodology

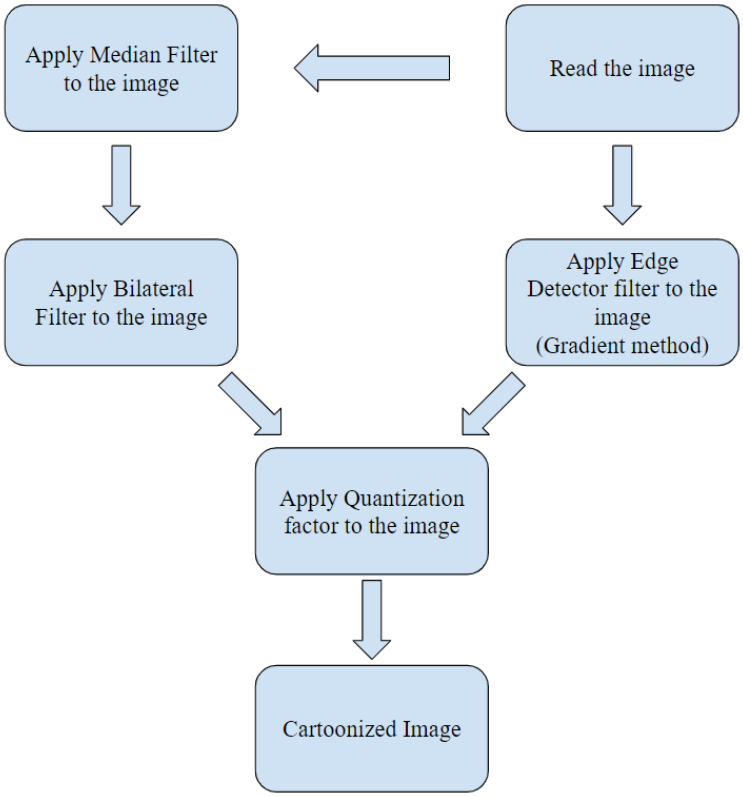


Fig 1: Methodology of the model

Algorithms

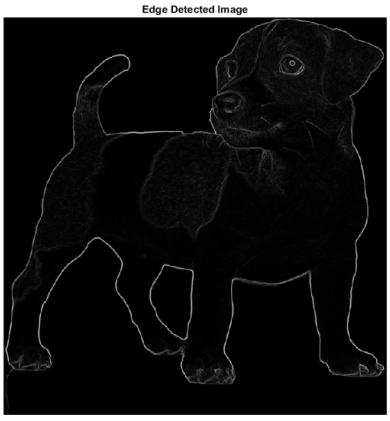
- *Read the image*: The image is read using the "imread" function of MATLAB. The image is stored as a matrix with values corresponding to the pixels of the image.
- *Median Filter*: This Filter takes a [m,n] matrix around each pixel of the image and sorts the matrix. The median value of the sorted matrix is given to the pixel at hand.
- *Bilateral Filter*: Biased Gaussian smoothing function is used, biased on the weighted brightness of the neighboring pixels. Smoothens the image without blurring it.
- *Edge Detector*: The Edge Detector uses the Gradient Edge Detection Method. Detects instant changes in the intensity (brightness) of the pixels of the image.

- *Quantizing the matrix*: Takes the image output of the Bilateral function and Edge Detector function. Applies quantization factor to discretise the matrix for easy operation.

Original Image

Median Filtered Image

Bilateral Filtered Image



Edge Detected Image

Quantized Image - Cartoonized Image

Steps to cartoonize images

- *Removing noise from the image:*
  In this step, the "salt and pepper" noise from the image is removed using filtering. There are two methods to filter the image, namely, Gaussian Filtering and Median Filtering. We use a Median filter as Gaussian filter deteriorates the image quality.
- *Smoothing the image:*
  The image is smoothened without distorting the image and ensuring the quality of the image. Biased Gaussian Bilateral Filtering is used for the same.
- *Edge Detection:*
  Edges of the image are detected by analyzing the pixel-wise sharp change in the intensity of the image.
- *Cartoonize images:*
  The image matrix is quantized and printed to cartoonize the image.

Removing noise from the image

- *Gaussian Filtering:* One of the most basic methods to remove noise from images. Diminishes the noise but doesn't actually remove the noise. Due to this, important details of the image are lost. As shown in the diagram, after applying the Gaussian Filter, the image gets blurred leading to the loss of the details of the coins.

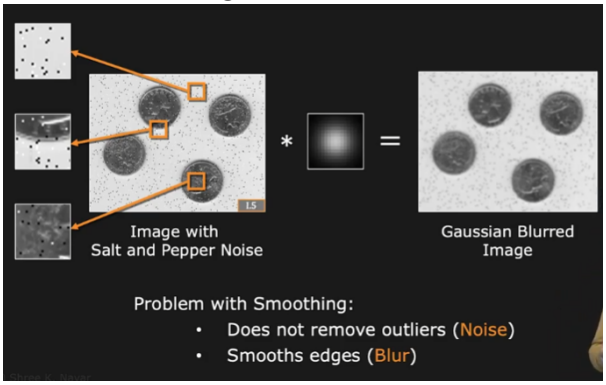
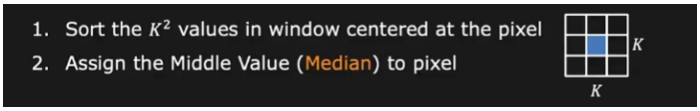


Fig 2: Gaussian Filtering

- *Median Filtering:* It is more of an algorithmic approach. In this method, a [k,k] matrix is taken around each pixel. This matrix is sorted and the middle value, the median, of the matrix is assigned to that particular pixel. The diagram shows that even with a small filter, the noise is effectively removed from the image without losing the details of the coins. Applying a larger median filter gives negative results. Larger filters tend to remove more noise but at the cost of the details of the image.



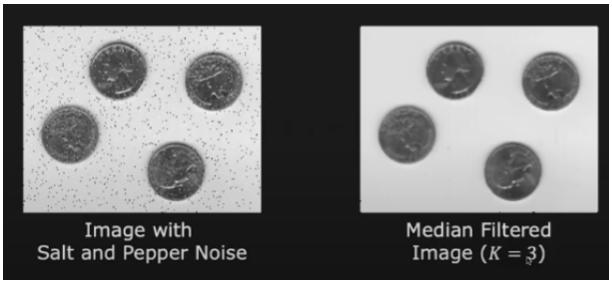


Fig 3: Median Filtering

Smoothing the image

- *Bilateral-Gaussian Smoothing:* Bilateral from the name suggests that two concepts are incorporated into one. Here, the spatial gaussian is applied to the image to smoothen the image. The image shown depicts the output that we get after applying a spatial gaussian filter. We can see that the image gets distorted and the edge of the image is lost.

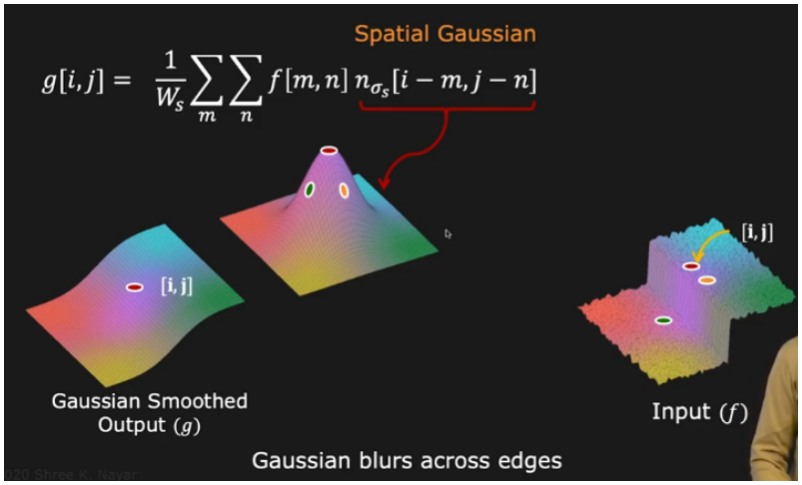


Fig 4: Bilateral-Gaussian Smoothing

- *Biased Spatial Gaussian Smoothing:* The spatial gaussian is combined with the brightness gaussian. In this, the gaussian smoothing applies to each pixel of the image also by comparing the brightness of the pixel to that of the neighboring pixels. Smaller brightness differences are given larger weight and larger brightness differences are given smaller weight. As shown in the image, the combined kernel takes care of the weight assigned to each pixel and hence the image at hand is not distorted. This method adapts the filter based on the parameters considered around the pixel, hence gives better results.

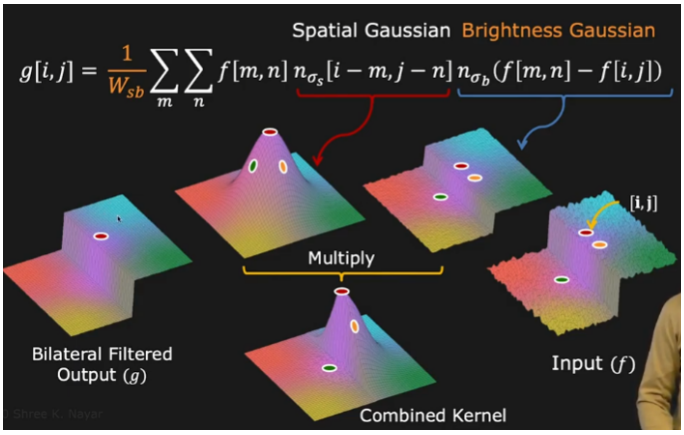


Fig 5: Biased Spatial Gaussian Smoothing

- *Weighting the pixels for brightness:* It is important to weight the brightness of the neighboring pixels as the energy of the image is to be kept constant at 1. As the weight of each pixel is different, it has to be changed for every pixel and hence it is given by the formula given in the image. Weights are calculated by summing the product of the brightness and spatial gaussians.

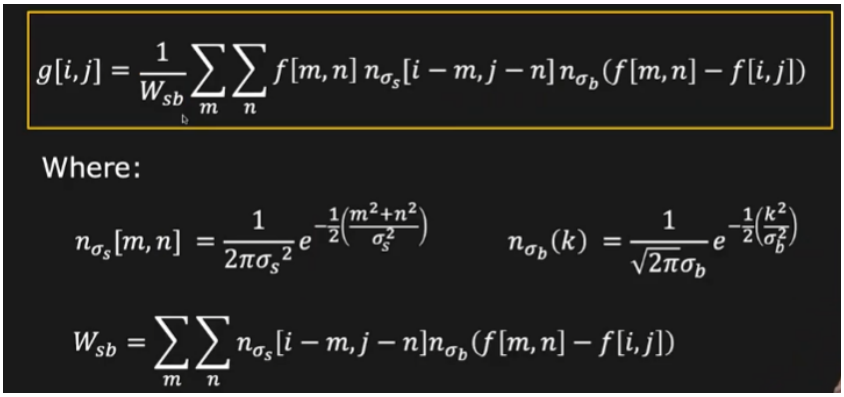


Fig 6: Mathematical Equation for different Smoothing

Edge Detection

- *Gradient method :* This method makes use of partial derivatives being applied to the change in brightness of the images in both horizontal and vertical directions. Image partial derivatives are taken using the gradient method ("Del operator"). The brightness value that is to be assigned to each pixel is calculated from the magnitude of the gradient which is shown in the given image. This detects edges in the images as edges are nothing but sharp brightness changes in any image.

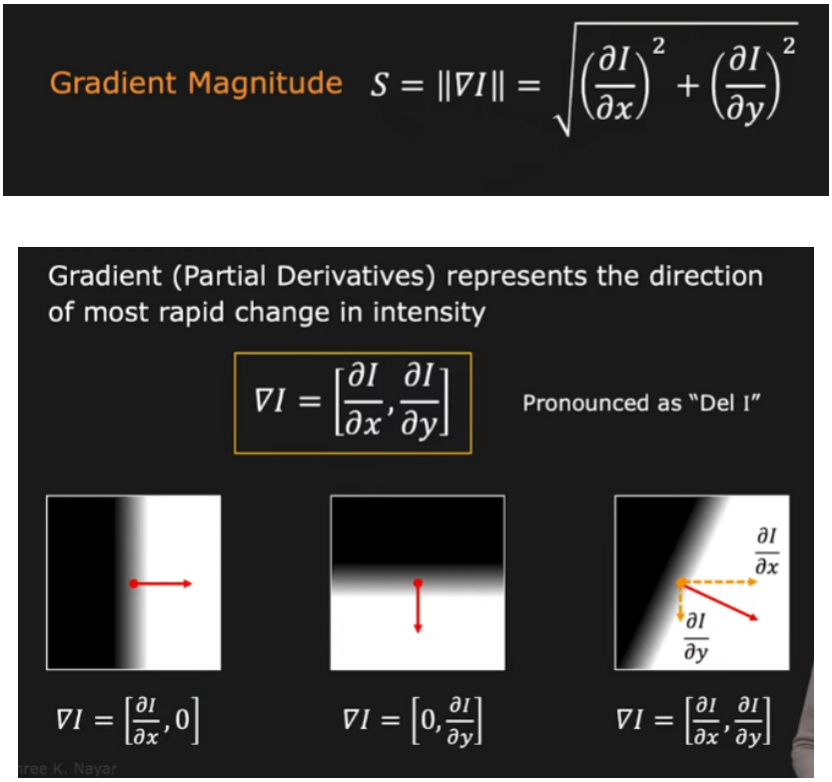


Fig 7: Mathematical Equation for gradient edge detection.

Quantized Image

After applying the Median Filter and the Biased Gaussian (Bilateral Filter), we get the final image matrix. This matrix is then discretized in the form of quanta to cartoonify the images. What it actually does is to assign a particular quanta value to a certain brightness range in the image. This quantization of the brightness gives the image a cartoon-like appearance.

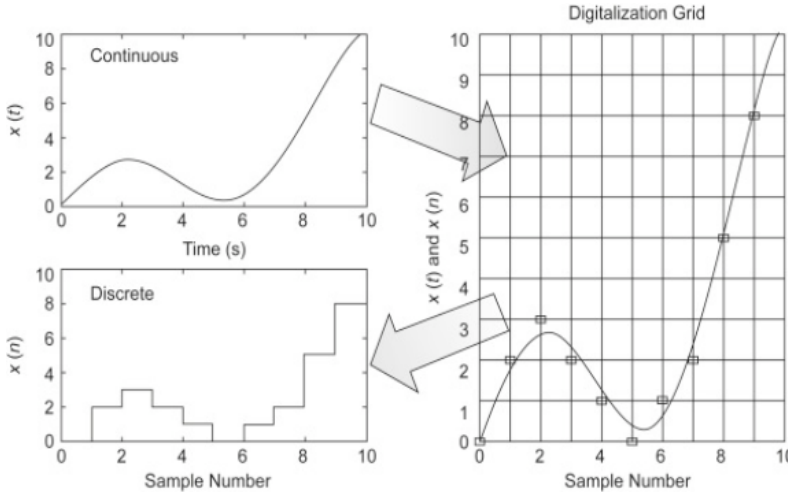
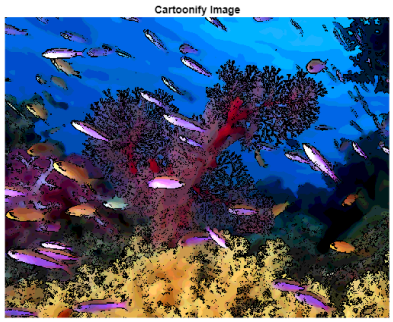
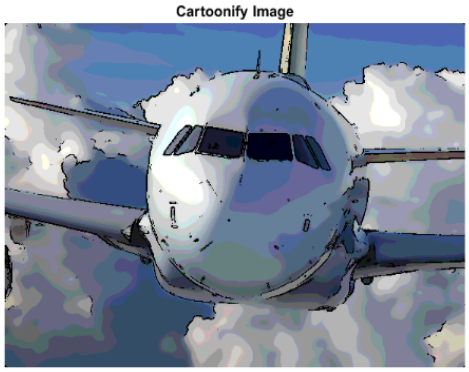


Fig 8: Quantized the image for Cartoonify the image.

Results





Conclusion

As with any model that seeks to meet an aesthetic goal, measuring the success of cartoonify images poses some difficult problems. Specific goals throughout the process were to achieve solid continuous contours while avoiding small line clutter in the image, and also to achieve a large region of homogeneous color.

References

[1]https://stacks.stanford.edu/file/druid:yt916dh6570/Dade_Toonify.pdf

[2]https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf

[3]https://www.youtube.com/watch?v=lOEBsQodtEQ&list=PL2zRqk16wsdqXEMpHrc4Qnb5rA1Cylrhx&index=3

[4]https://www.youtube.com/watch?v=7FP7ndMEfsc