

Basic Image processing techniques through R-Language

¹B.Rajesh, ²C.Reddi Neelima, ³V.Rajya Lakshmi, ⁴M.S. Vani, ⁵N.V.Neeraja,

Assistant Professor
Computer Science and Engineering,
Mother Theresa Institute of Engineering and Technology, Palamaner, India

Abstract: Image processing is a method to analysis and manipulation of a digitized image, especially in order to improve its quality, it plays an important role in many streams like medical, security, banking and all type of criminal activities. To days world generates huge amount of image database and it increase day by day massively so the tools like MATLAB is not sufficient to compute and analyze that much of data, in this paper discussing about such data handling tool namely R. R is a language as well as perfect tool to analyze big data, but till now R is not proven that it can also handle image processing. I want to prove R tool in image processing can handle the huge amount of data, In R-language there are mainly three packages (magick, imager, EBImage) that hold the basic image processing techniques like transformations, filters, color management, histogram equalization and so on Shown with the R tool. For processing an image by using R-language use R studio, we can take any real time picture as image as an input in this project. Finally we can shows R is a powerful tool to processing an image compared to MATLAB.

Index Terms: Image Processing, R tool, magick, imager, EBImage, transformations, filters, color management, histogram.

I. INTRODUCTION

Image Processing is not a new word in present days many researchers have been taken place in this platform. Image processing is the technique to covert image graphics and digital form and performs some mathematical operation by using images or video such as a photograph or video frame. It is a type of signal processing in which input is an image and output may be image characteristics. Nowadays, image processing is among rapidly growing technologies. Basically analog and digital are the techniques in image processing. At present digital is the method using in image processing because of its advantages. In image analysis many type tools are used like mat lab, Hadoop[1], python etc. To enhance the image we have operations like transformations, filters, gamma correction, cropping, brightness and contrast, color management. Each method has their own method of implementation to enhance the given input image. In this paper we are going to explain how these methods are going to be implemented in R tool. At starting we take any real time image as input and then read image and write it to another then display it, after check properties of image.

Once image is selected we perform number of image processing techniques initially we start with checking of brightness and adjust contrast in image. Then we apply number of transformations and rotations on that, once is there any one image existing noise to remove that by using filters, we have two filters namely LOW PASS and HIGH PASS and finally result shown by using histogram Equalization.

R-LANGUAGE

R is a one type of statistical programming language. Which can provide various techniques for image processing like transformations, filters, gamma correction, cropping, brightness and contrast, color management etc., "R environment run based on CRAN mirror. it is a collection of packages and tools building to Run R-programs in a R-studio." R is a combining process of facilities for image data modification, calculation and display it mainly includes an effective data storage and data handling facility. R allows the user to add different operations by defining new operations. The list of packages that are used to implement image processing techniques in R are magick, imager, EBImage etc..

Then the remaining paper is organized as section 2, section 3, and section 4, in section 5 we examine related work, proposed work, and experimental result and finally conclude the paper.

II. RELATED WORK

Yun Zhang has clearly explained about image gamma correction [2] importance, techniques and limitations. A survey paper by Ebtsam Adel et.al [3] explains about the image feature extraction techniques. Image enhancement using fusion techniques is explained in [4] by Anju Rani. Several works are done in respective area of research using mat lab[5] and complete implementation work is stated in [6] using Raspberry Pi. Methods to improve image enhancement based on image fusion is explained by Fang et.al [7]. Magick[11] imager is a package gives techniques to process image processing and also there is an another is an another package called EBImage, it is also good package in R language to make image processing[12].

III. PROPOSED SOLUTION

Due to the disadvantages described in the existing system ie., MATLAB does not support large amount of image data to process so to overcome this we introduce R-tool to do image processing in a well bounded manner using some packages that are related to

image processing. Packages like (magick, imager,EBImage) that hold the basic image processing techniques like transformations, filters, animations etc... Also the Two other Sample operations such as image stitching and fusion are shown with the R tool.

Image processing techniques:

The various techniques that are used to process images are

1. Load image
2. Displaying image
3. Check properties
4. Adjusting Brightness, Contrast and Gamma correction
5. Spatial Transformation
6. Color Management
7. Image Filtering
8. Histogram Equalization

First we load an image as an input and then Image can be displayed using display function and pixel intensities should range from 0 (black) to 1 (white) then check properties of the image is there any adjustments do by using some techniques like Brightness, contrast and Gamma correction. Spatial transformations are used to make image in different formats, image cantoning any noise by using filters we remove then corrections are done the image equalization visualize in the form of Histogram.

IV. SYSTEM DESIGN

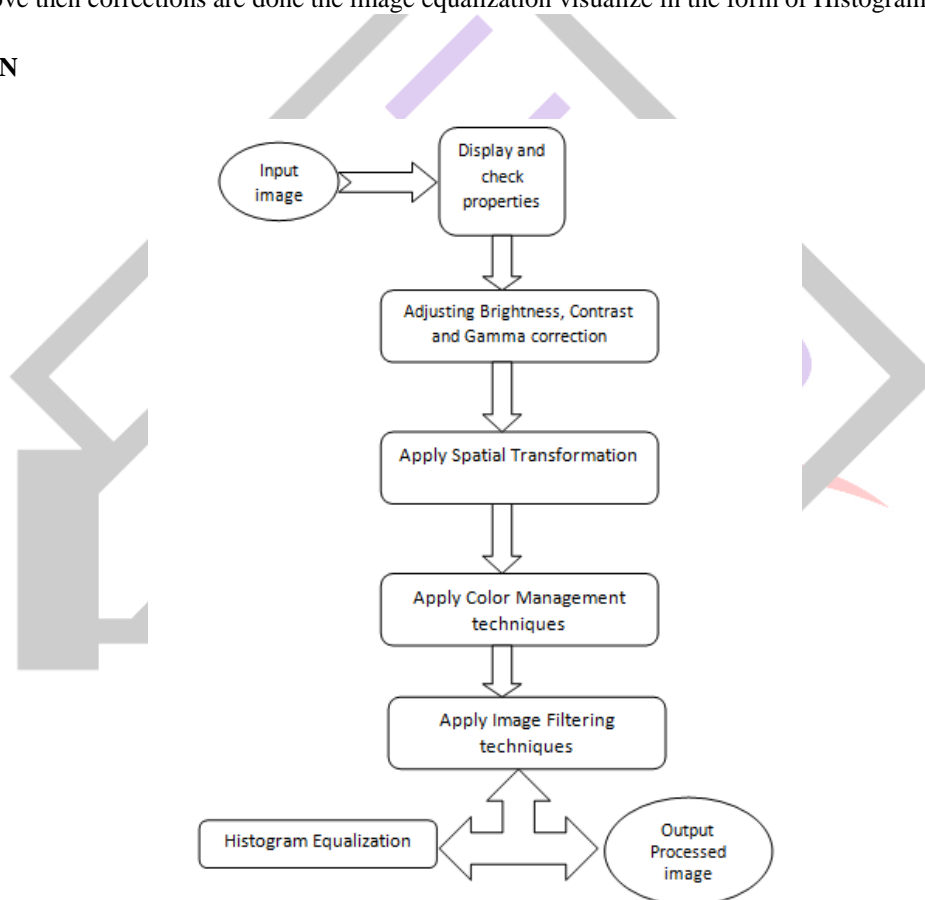


Fig.2. System Architecture

V. EXPERIMENTAL RESULTS

Performing Image processing and analysis in R, installation of EBImage package is required. Execute the below code to get package installed in R.

```

source("http://bioconductor.org/biocLite.R")
biocLite()
biocLite("EBImage")
  
```

Performing the Reading, Writing and displaying of Image

The loading of package will be done by using the below command.

```
library (EBImage)
```

Reading Image: The readImage function is used to read Image from file location or URLs

```
Image <- readImage("C:\\Users\\Public\\Pictures\\Sample Pictures\\ Penguins.jpg")
```

Displaying Image: Image can be displayed using display function and pixel intensities should range from 0 (black) to 1 (white)

```
display (Image)
```



Writing Image: WriteImage function is used to write the Image at a particular location.

```
writeImage(Image, 'E:\\imgc.jpeg' ,quality=85)
```

Image Properties

Images are stored as multi-dimensional arrays containing the pixel intensities, so if we use the print function it will return the image properties. In the below output two section is present i.e. summary and array of pixels.

```
>print(Image)
Image
  colorMode      : Color
  storage.mode   : double
  dim            : 1024 768 3
  frames.total   : 3
  frames.render  : 1

imageData(object)[1:5,1:6,1]
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.4549020 0.4549020 0.4627451 0.4588235 0.4666667 0.4745098
[2,] 0.4627451 0.4549020 0.4588235 0.4588235 0.4627451 0.4666667
[3,] 0.4627451 0.4588235 0.4588235 0.4627451 0.4588235 0.4588235
[4,] 0.4705882 0.4627451 0.4627451 0.4627451 0.4627451 0.4627451
[5,] 0.4627451 0.4549020 0.4627451 0.4627451 0.4627451 0.4666667
```

Image matrices operation for Adjusting Brightness, Contrast and Gamma correction:

| IMAGE SITTING | MATHEMATICAL OPERATORS |
|----------------------|------------------------|
| Adjusting Brightness | + or - |
| Adjusting contrast | * |
| Gamma correction | ^ |

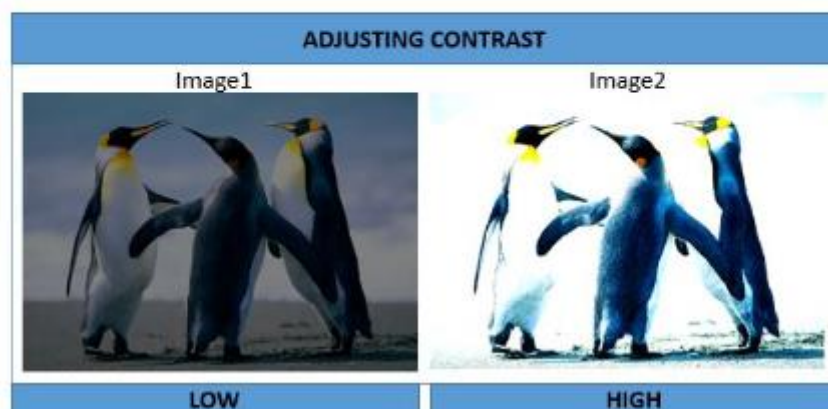
Adjusting Brightness

```
Image1 <- Image + 0.2
Image2 <- Image - 0.2
display (Image1); display(Image2)
```



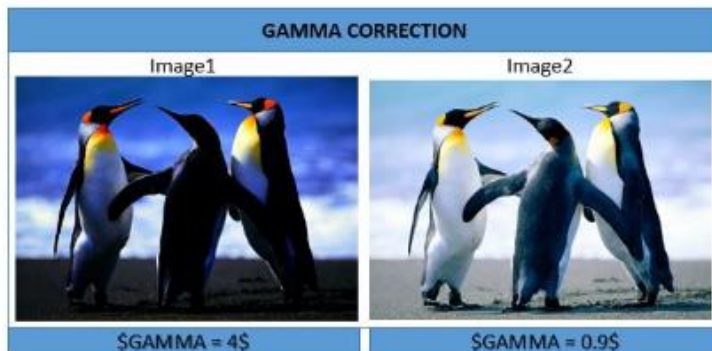
Adjusting Contrast

```
Image1 <- Image * 0.5
Image2 <- Image * 2
display(Image1); display(Image2)
```



Gamma Correction

```
Image1 <- Image ^ 4
Image2 <- Image ^ 0.9
display(Image1); display(Image2)
```



Cropping Image

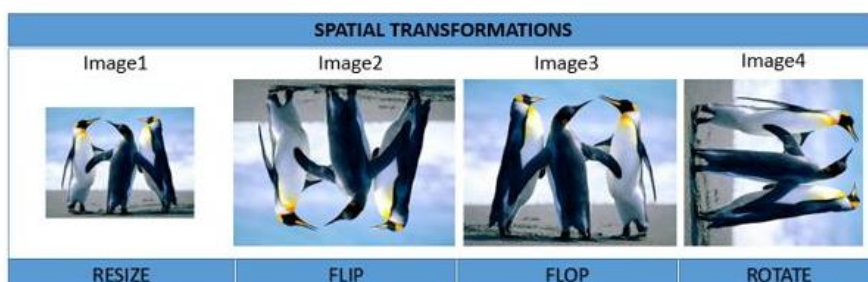
```
Image1 <- Image[1:800, 1:200, ]
display(Image1)
```



Spatial Transformation

Spatial image transformations can be performed with the functions such as re-size, rotate, translate and the functions flip and flop to reflect images.

```
#Resizing the image
Image1 <- resize(Image, 500)
display(Image1)
# Flip and Flop of image
Image2 <- flip(Image)
display(Image2)
Image3 = flop(Image)
display(Image3)
#Image Rotation
Image4 <- translate(rotate(Image, 90), c(50, 0))
display(Image4)
```



Color Management

The function `colorMode` can access and change the value of the slot `colormode`, altering the rendering mode of an image. In this, the Color image (Image) with one frame is changed into a Grayscale, corresponding to the red, green and blue channels.

```
>print(Image)
Image
  colorMode      : Color
  storage.mode   : double
  dim            : 1024 768 3
  frames.total   : 3
  frames.render  : 1

imageData(object)[1:5,1:6,1]
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.4549020 0.4549020 0.4627451 0.4588235 0.4666667 0.4745098
[2,] 0.4627451 0.4549020 0.4588235 0.4588235 0.4627451 0.4666667
[3,] 0.4627451 0.4588235 0.4588235 0.4627451 0.4588235 0.4588235
[4,] 0.4705882 0.4627451 0.4627451 0.4627451 0.4627451 0.4627451
[5,] 0.4627451 0.4549020 0.4627451 0.4627451 0.4627451 0.4666667

>#Greyscale Image
>colorMode(Image) <- Grayscale
>display(Image)
```

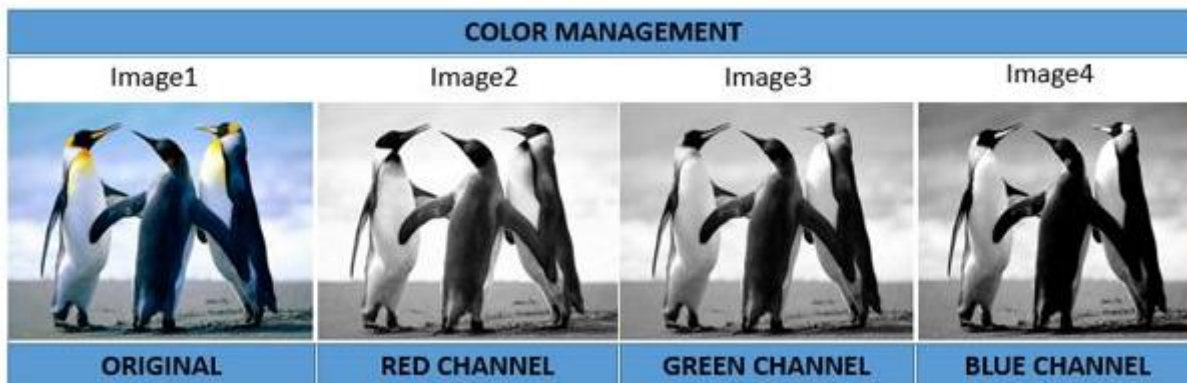
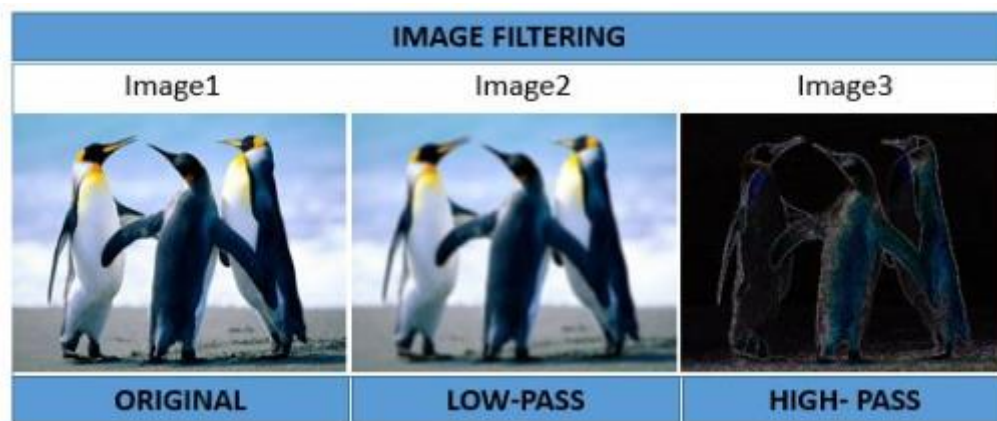


Image Filtering

Images can be linearly filtered using `filter2`. **Low-pass filtering** is used to perform blur image, remove noise, etc. **High-pass filtering** is used to perform detect edges, sharpen images etc., various filter shapes can be generated using `makeBrush`.

```
#LOW-PASS Filtering image
flo = makeBrush(21, shape='disc', step=FALSE)^2
flo = flo/sum(flo)
imgflo = filter2(Image, flo)
display(imgflo)
#HIGH-PASS Filtering image
fhi = matrix(1, nc=3, nr=3)
fhi[2,2] = -8
imgfhi = filter2(Image, fhi)
display(imgfhi)
```

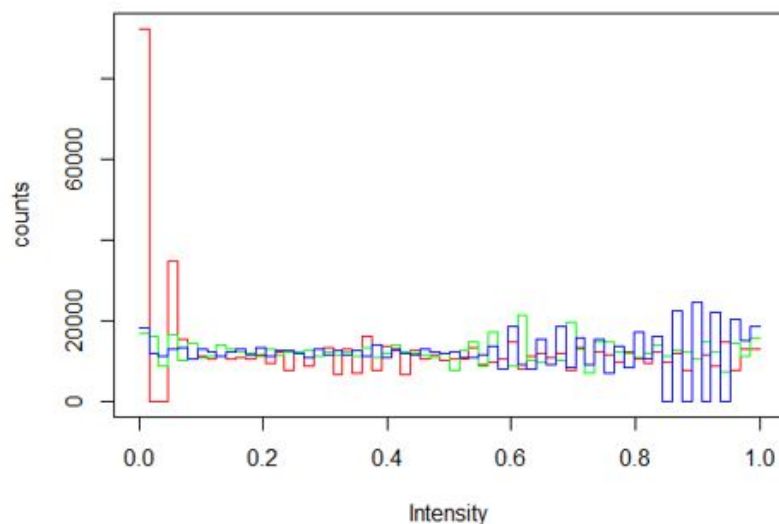


Histogram Equalization

Equalize the image histogram to a specified range and number of levels. Individual channels of Color images and frames of image stacks are equalized separately.

```
y= equalize(Image) hist(y)
```

Image histogram: 2359296 pixels



VI. CONCLUSION

When compared to traditional tools R is a powerful tool for image processing. Thus, in this paper we are showing results for image processing methods like Load image, Displaying image, Check properties, Adjusting Brightness, Contrast and Gamma correction, Spatial Transformation, Color Management, Image Filtering and Histogram Equalization using R Studio. Further implementations like image edge detection, mosaic, image enhancement etc. can also be implemented in R.

REFERENCES

- [1] T. P. Reddy and V. S. A. Setti, "A Study on Median Filter Using Big Data," International Journal of Advanced research in Computers and communication Engineering, vol. 5, no. 10, 2016.
- [2] Y. Zhang, Understanding Image Fusion, 2004.
- [3] E. Adel, M. Elmogly and H. Elbakry, "Image Stitching based on Feature Extraction Techniques:A survey," International Journal of Computer Applications, vol. 99, no. 6, 2014.
- [4] A. Rani and G. Kaur, "Image Enhancement using Image Fusion Techniques," Internation Journal of Advanced Reasearch in Computer Science and Software Engineering, vol. 4, no. 9, 2014.
- [5] T. Patil, S. Mishra, P. Chaudhari and S. Khandale, "Image Stitching using Matlab," Internation Journal of Engineering of Trends and Technology , vol. 4, no. 3, 2013.

- [6] K. shree, Lokesha.H and H. S. kumar, "Implementation of Image Processing on Raspberry Pi," International Journal of advanced Research in Computer and Communication Engineering, vol. 4, no. 5, 2015.
- [7] X. Fang, J. Liu, W. Gu and Y. Tang, "A Method to improve the image enhancement result based on image fusion," International Conference on Multimedia Technology, pp. 55-58, 2011.
- [8] P. Kale and K.R.Singh, "A Technical Analysis of Image Stitching Algorithm," International Journal of Computer Science and Information Technologies, vol. 6, no. 1, 2015.
- [9] D. K. Sahu and M.P.Parsai, "Different Image Fusion Techniques-A Critical Review," International Journal of Modern Engineering and Research, vol. 2, no. 5, 2012.
- [10] P. M. Jain and V. K. Shandliya, "A Review paper on Various approaches of Image Mosacing," International Journal of Computational Engineering and Research, vol. 3, no. 4, 2013.
- [11] <http://blog.thedigitalgroup.com/rajendras/2015/06/12/imageprocessing>.
- [12] <http://www.bioconductor.org/packages/release/bioc/manuals/EBImage/man/EBImage>
- [13] <https://www.r-bloggers.com/r-image-analysis-using-ebimage/>

