



## **Efficient Image Enhancement Techniques**

**Muhammad Shahzad\***  
**Shiraz Latif\***

*Department of Computer Science  
Usman Institute of Technology, Karachi, Pakistan*

**Quratulain Akhter\***  
*Department of Software Engineering  
MAJU, Karachi, Pakistan*

**Farida Bibi\***  
*Department of Computer Engineering  
N.E.D UET, Karachi, Pakistan*

### **ABSTRACT**

Image enhancement, one of the categories of image processing is being used worldwide as a technique to manipulate images in a way to make them worthy for a particular use. Many of the fields like Medical, textile, GIS etc are being fully benefited by the use of such techniques. In this paper, we explore various image enhancements techniques (like image addition, subtraction, inversion, convolution, noise removal etc.) with the help of .NET framework libraries. It is concluded after experiments that Bitmap and BitmapData classes of .NET framework allow direct memory access and make image processing much faster and efficient.

**INSPEC Classification :** C1250,C4240, C5260

**Keywords :** image processing, image enhancement, Bitmap, .NET framework

### **1. INTRODUCTION**

Holding its major role in various areas of life, the field of image processing has been under research for years for the efficient and fast processing of images. The experimental results of these researches keep on adding new tools (Atul, Rochak, Saini, 2007) and softwares which help in lifting the performance graph of image processing day- by-day. The .NET framework of Microsoft provides various libraries as built-in classes for easy and fast image processing.

Bitmap class of .NET framework provides quicker byte-level memory access which reduces the extra time in accessing each pixel of an image for processing. This paper is organized as follows. Section 2 is based on theoretical description of the library used for image processing. Some experimental results are also shown in this section. Section 3 compares

\* The material presented by the authors does not necessarily portray the viewpoint of the editors and the management of the Institute of Business and Technology (BIZTEK) or Usman Institute of Technology, MAJU and N.E.D UET, Karachi, Pakistan.

\* Muhammad Shahzad :mshahzad@uit.edu

\* Shiraz Latif : m\_nickpour@yahoo.com

\* Quratulain Akhter :quratulain.akhter@gmail.com

\* Farida Bibi : farida.salam@gmail.com

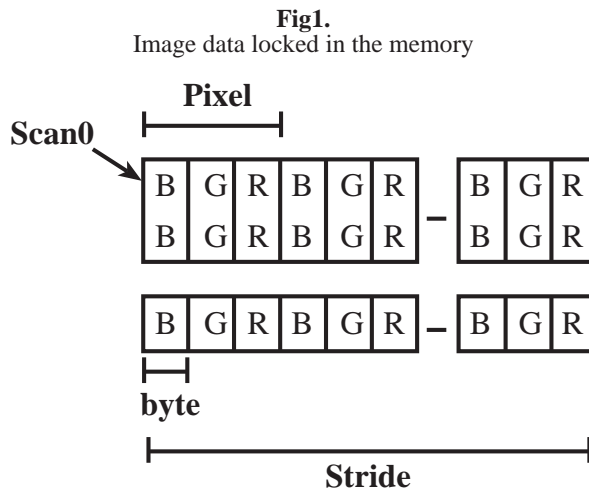
© JICT is published by the Institute of Business and Technology (BIZTEK).  
Ibrahim Hydri Road, Korangi Creek, Karachi-75190, Pakistan.

the efficiency of method proposed with other methods available for image processing. Rest of the sections present the conclusion, acknowledgements, references and brief introduction of the authors of the paper.

## 2. .NET FRAMEWORK LIBRARIES FOR IMAGE PROCESSING

.NET framework provides two fantastic classes i.e. Bitmap and BitmapData for image processing. Bitmap class locks the image to be processed in the memory and this locking helps in the direct byte-level access of the pixels of an image. This locking is achieved by using LockBits method of Bitmap class which enables to fix a portion of the bitmap pixel data array in memory, access it directly and finally replace the bits in the bitmap with the modified data. Lock Bits returns a Bitmap Data class that describes the layout and position of the data in the locked array ([Http://www.msdn.microsoft.com](http://www.msdn.microsoft.com)).

The data can be manipulated pixel-wise using pointers. Depending on the pixel format (number of bytes associated with each pixel) of the image, each pixel can be accessed using scan0 (indicates the address of the first byte of first pixel) (<http://www.bobpowell.net/lockingbits.htm>) and stride (<http://www.bobpowell.net/lockingbits.htm>), (The width, in bytes, of a single row of pixel data) properties of Bitmap class



Using this class, an image manipulation becomes as easy as never before. The time, which most of the softwares spend in reading/writing the pixel value, reduces immensely.

The process of the system proposed is as follows. The system first converts the input images (gif, jpeg etc) into bitmap. The bytes of the newly converted bitmap image are then locked into the memory using LockBits method of Bitmap class.

```
BitmapData bmpData =  
    bmpSourceImage.LockBits(  
        new Rectangle(0, 0, bmpSourceImage.Width, bmpSourceImage.Height),  
        ImageLockMode.ReadWrite,  
        PixelFormat.Format24bppRgb);
```

A byte pointer is then initialised with the starting address of the image in the memory with Scan0 property of Bitmap class.

```
byte*bytePointerToImageAddress=(byte*)bmpSourceImage.Scan();
```

Now, each byte of any pixel is accessible using the byte pointer and can be manipulated according to the respective operation of image processing.

```
unsafe
{
    for(
        int intRowIndex=0;
        intRowIndex<bmpSourceImage.Hight;
        intRowIndex++
    )
    {
        for(
            int intColumnIndex=0;
            intColumnIndex<bmpSourceImage.Width*3;
            intColumnIndex++
        )
        {
            bytePointerToImageAddress[0]=
                (byteX225-bytePointerToImageAddress[0]);
            bytePointerToImageAddress++;
        }
    }
}
```

We did experiment with various image enhancement operations using these classes

### A. Image Addition

The addition of two images is performed straightforwardly in a single pass. The output pixel values are given by:

$$Y(i,j)=X_1(i,j)+ X_2(i,j)$$

Where  $X_1$  and  $X_2$  are two images to be added together and  $Y$  is the resultant image obtained.  $i$  and  $j$  represent the location of particular pixel in the image

### B. Image Subtraction

The subtraction of two images is given by:

$$Y(i,j)=X_1(i,j)- X_2(i,j)$$

Where  $X_1$  and  $X_2$  are two images for subtraction and  $Y$  is the resultant image obtained.  $i$  and  $j$  represent the location of particular pixel in the image

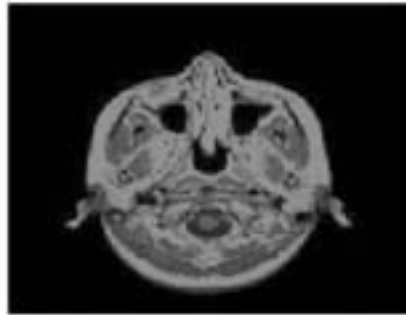
### C. Image Inversion

An inversion of an image is given by:

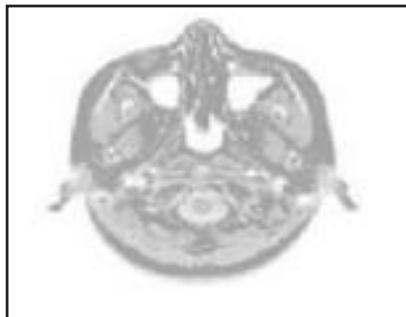
$$Y(i,j)=\text{max\_value} - X(i,j)$$

Where X is an image to be inverted. Y is the output image obtained, *max\_value* is the maximum colour value assigned to the pixel. *i* and *j* represent the location of particular pixel in the image

**Fig 2**  
Original Image



**Fig 3**  
Inverted Image



#### **D. Image Convolution**

Image convolution operation (also known as image filter) is used widely in the field of image processing for adding special effects (like blurring, edge sharpening etc) to the images. Mathematically, Convolution is defined as:

$$Y(i,j) = I(i,j) * K(i,j)$$

$$Y(i,j) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} I(i-k,j-l) * K(k,l)$$

Where I is an image to be convolved with K. K is the rectangular matrix with few integers (<http://www.codeproject.com/KB/graphics/ImageConvolution.aspx>). Y is the output image obtained. Changing the values of the integers in Kernel would add different effects to the image (<http://www.eas.asu.edu/~karam/2dconvolution/>). We implemented this operation by changing different values of the kernel as shown below

### i. Blurring the Image

**Fig 4.**  
Image 1



**Fig 5.**  
Image 2



### ii. Sharpen the blurred image

**Fig 6.**  
Image 1



**Fig 7.**  
Image 2



The magic of convolution lies in the integer values of the kernel. Changing the kernel value would change the result accordingly.

### 3. COMPARISON OF METHODS FOR CONVOLUTION

Normally, in spatial domain, the convolution performed is slower as compared to frequency domain. Matlab also requires image to be first converted into frequency domain before convolving the image. But .NET not only speeds up this operation but also make it easier to implement which other tools do not.

### 4. CONCLUSION

After thorough study and experimentations, it is concluded that Bitmap and Bitmap Data classes of .NET framework not only speeds up the operation of image processing but make it pretty easier to implement these operations.

The direction of our future work should include the implementation of more complex operations like image compression, image restoration, image resizing etc.

### **REFERENCES**

- Atul, Rochak, Saini "Simulation of Image Enhancement Techniques using MATLAB,"  
Proceedings of the First Asia International Conference on Modelling & Simulation,  
pp. 2007 Page(s): 296- 301.  
[Http://www.msdn.microsoft.com](http://www.msdn.microsoft.com)  
<http://www.eas.asu.edu/~karam/2dconvolution/>  
<http://www.bobpowell.net/lockingbits.htm>  
<http://www.codeproject.com/KB/graphics/ImageConvolution.aspx>