

# PROJECT REPORT

## Assignment-1a

CLID: sxb8547

Email-id:saisreebhagi@gmail.com

1. Write an openCV program that does the following – Read in a color image – Prompt the user for a (row, column) address – Check to make sure that (row, column) is inside the image and not on the image border – Call a subroutine “readBlock” as follows void readBlock(IplImage\* image, int row, int column); Print out a 3 by 3 block of pixel values centered at (row, column) in “image”

### Solution:

#### Source code:

```
#include <opencv/cv.h>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

void readBlock(IplImage *img, int row, int column);

int main(int argc, char *argv)
{
    //Loading an image in opencv

    IplImage *img=cvLoadImage("def.jpg",1);

    // checking whether color image is loaded in .vcproj "sxb8547"
```

```

if(!img)
{
    printf("Error:Can't open the file \n");
    return 0;
}

int row,column;

// Prompting the user for a row,column values of a pixel

printf("Enter row value and column values for the pixel \n");
printf("\n");
scanf("%d%d",&row,&column);

//code for checking whether pixel is within the image border

if((row<=img->height&&row>0)&&(column>0 && column<=img->width))

{
    int i=0,j=0;
    printf("\n The row and column values of a pixel are inside the loaded image \n");
    printf("\t\n");

    printf("BGR values of a pixel in matrix form \n");

    // condition when a pixel is anywhere in the first horizontal line of 3*3 block

    if(i==0&&j<=img->width)

    {

```

```

        if(row==i&&column==j)
        {
            printf("\n In border");

        }
    }
    else

```

vertical line of 3\*3 block //condition when a pixel is anywhere in the first

```

        if(i<=img->height&&j==0)
        {
            if(row==i&&column==j)
            {
                printf("\n In border");

            }
        }
    }

```

of 3\*3 block //condition when pixel is anywhere at the last horizontal line

```

    else if(i==img->height&&j<=img->width)
    {
        if(row==i&&column==j)
        {
            printf("\n In border");

        }
    }

```

```

    }

    // condition when a pixel is anywhere at the last vertical
line of 3*3 block

    else if(i<=img->height&& j==img->width)
    {
        if(row==i&&column==j)
        {
            printf("\n In border");
        }
    }

    readBlock(img,row,column);

}

    getch();
}

//code for printing 3*3 block of BGR values of a pixels surrounded by a given pixel's
row,column values
void readBlock(IplImage *img,int row,int column)

{

    char* thisrow;
    uchar pixelval;
    for(int l=row-1;l<row+2;l++)
    {

```

```

        thisrow=(img->imageData+l*(img->widthStep));
        printf("\n");

    for(int m=column-1;m<column+2;m++)
    {
        for(int channel=0;channel<img->nChannels;channel++)
        {

            pixelval = thisrow[(img->nChannels)*m + channel];
            printf("%d ",pixelval);

        }
        printf("\t");
    }
}

```

### Input:

After loading a color image "def.jpg",user enters a row and column values of a pixel shown as below:

**When row,column values of a pixel are inside the loaded image:**

**I/p:**

**1)Enter row value and column values for the pixel**

**100**

**300**

### Output:

**The row and column values of a pixel are inside the loaded image**

**BGR values of a pixel in matrix form**

**251 240 218    251 240 218    251 240 218**

**252 239 217    252 239 217    252 239 217**

**252 239 217    252 239 217    252 239 217**

**Output Screenshot:**



C:\Users\saisree\Desktop\Visual Studio 2010\sxb8547\

Enter row value and column values for the pixel

100

300

The row and column values of a pixel are inside th

BGR values of a pixel in matrix form

251	240	218	251	240	218	251	240	218
252	239	217	252	239	217	252	239	217
252	239	217	252	239	217	252	239	217



2)Enter row value and column values for the pixel

1000

1250

**Output:**

The row and column values of a pixel are inside the loaded image

BGR values of a pixel in matrix form

243 213 166    243 213 166    242 212 165

243 213 166    243 213 166    243 213 166

243 213 166    243 213 166    243 213 166

**Output screenshot:**





C:\Users\saisree\Desktop\Visual Studio 2010\sxb8547\Debug\sxb8547.exe

Enter row value and column values for the pixel

1000

1250

The row and column values of a pixel are inside the loaded image

BGR values of a pixel in matrix form

243	213	166	243	213	166	242	212	165
243	213	166	243	213	166	243	213	166
243	213	166	243	213	166	243	213	166



3) This is the error we get when we can't load any image in the Visual studio project "sxb8547"

Error:Can't open the file

Press any key to continue . . .

4)Enter row value and column values for the pixel

1100

1300

**Output:**

The row and column values of a pixel are inside the loaded image

BGR values of a pixel in matrix form

249 225 179    249 225 179    249 225 179

249 225 179    249 225 179    249 225 179

249 225 179    249 225 179    249 225 179

**Output screenshot:**



C:\Users\saisree\Desktop\Visual Studio 2010\sxb8547\Debug\sx

Enter row value and column values for the pixel

1100

1300

The row and column values of a pixel are inside the loaded

BGR values of a pixel in matrix form

249	225	179	249	225	179	249	225	179
249	225	179	249	225	179	249	225	179
249	225	179	249	225	179	249	225	179

\_



## Tools used:

1. Microsoft visual studio 10.0 (2010)
2. OpenCv 2.4.6

## Description:

In this project, I loaded a color image "def.jpg" by using `cvLoadImage()`. Here, I used `IplImage` structure to get all the properties of a color image.

After loading, I entered row, column values of a pixel for finding whether pixel is within or on the border of the image.

If a pixel value is within the loaded image, then I displayed 3\*3 block of BGR values surrounded by that pixel in matrix form by using the following formulae below in a `readBlock` function :

```
void readBlock(IplImage *img, int row, int column)
{
    char* thisrow;
    uchar pixelval;
    for(int l=row-1; l<row+2; l++)
    {
        thisrow=(img->imageData+l*(img->widthStep));
        printf("\n");
        for(int m=column-1; m<column+2; m++)
        {
            for(int channel=0; channel<img->nChannels; channel++)
            {
                pixelval = thisrow[(img->nChannels)*m + channel];
                printf("%d ", pixelval);
            }
            printf("\t");
        }
    }
}
```

}

}

}