

Q.1 CODE :

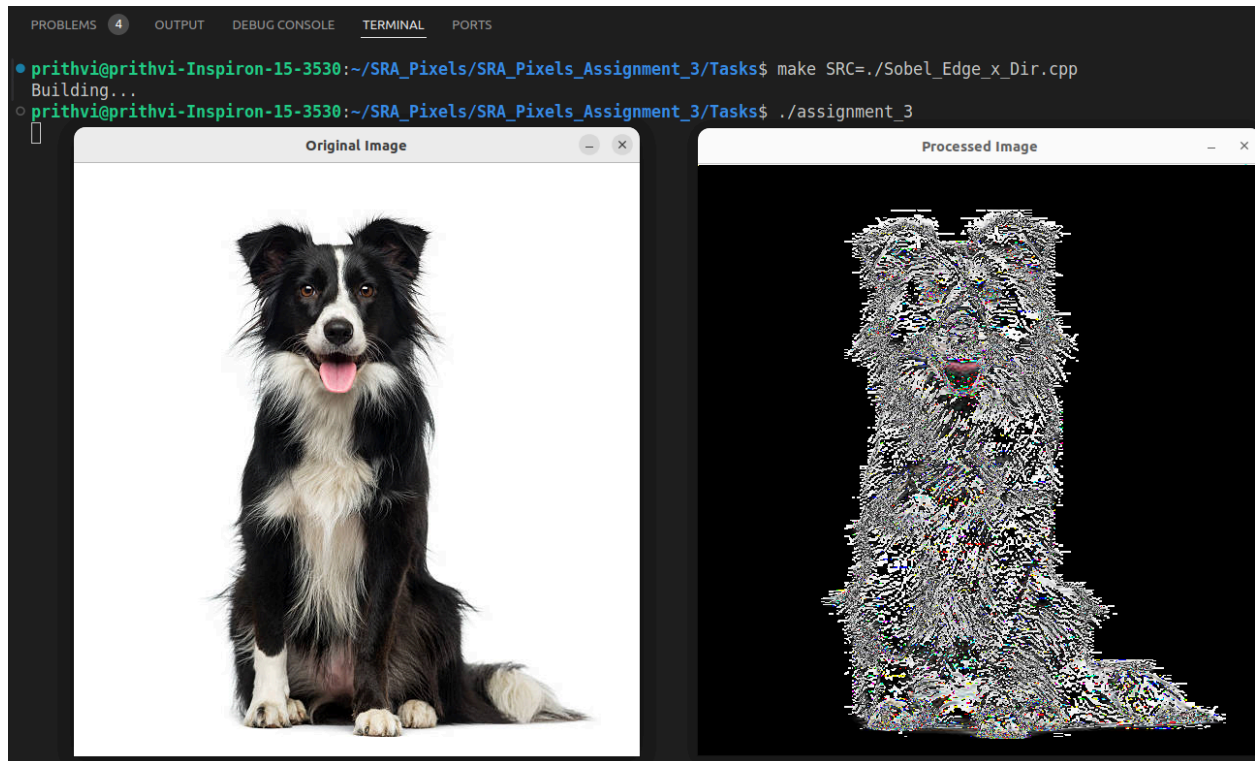
```
Sobel_Edge_x_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_x_Dir.cpp > ...
1  #include <iostream>
2
3  #include <opencv2/highgui.hpp>
4  #include <opencv2/imgcodecs.hpp>
5  #include <opencv2/imgproc.hpp>
6
7  using namespace std;
8  using namespace cv;
9
10 int main(){
11     Mat img;
12     img=imread("dog.jpg");
13     int img_wid=img.size().width;
14     int img_hgt=img.size().height;
15     Mat img1((img_hgt-2), (img_wid-2), CV_8UC3, Scalar(0, 0, 0));
16     imshow("Original Image", img);
17     for(int row=0; row<(img_hgt-2); row++){
18         for(int col=0; col<(img_wid-2); col++){
19             for(int i=0; i<3; i++){
20                 img1.at<Vec3b>(row, col)[i]=(((-1)*(img.at<Vec3b>((row-1), (col-1))[i]))+(0*(img.at<Vec3b>(row, (col-1)
21                 if(img1.at<Vec3b>(row, col)[i]>255){
22                     img1.at<Vec3b>(row, col)[i]=255;
23                 }
24             }
25         }
26     }
27     imshow("Processed Image", img1);
28     waitKey(0);
29     return 0;
30 }
```

```
Sobel_Edge_x_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_x_Dir.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 )+(1*(img.at<Vec3b>((row+1), (col-1))[i]))+((-2)*(img.at<Vec3b>((row-1), col)[i]))+(0*(img.at<Vec3b>(row, col)[i]))+(2*(img.at<Vec3b>((
21
22
23
24
25
26
27
28
29
30
```

```
Sobel_Edge_x_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_x_Dir.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 +(2*(img.at<Vec3b>((row+1), col)[i]))+((-1)*(img.at<Vec3b>((row-1), (col+1))[i]))+(0*(img.at<Vec3b>(row, (col+1))[i]))+(1*(img.at<Vec3b>
21
22
23
24
25
26
27
28
29
30
```

```
Sobel_Edge_x_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_x_Dir.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 ol)[i]))+((-1)*(img.at<Vec3b>((row-1), (col+1))[i]))+(0*(img.at<Vec3b>(row, (col+1))[i]))+(1*(img.at<Vec3b>((row+1), (col+1))[i])));
21
22
23
24
25
26
27
28
29
30
```

Q.1 OUTPUT :



Q.2 CODE :

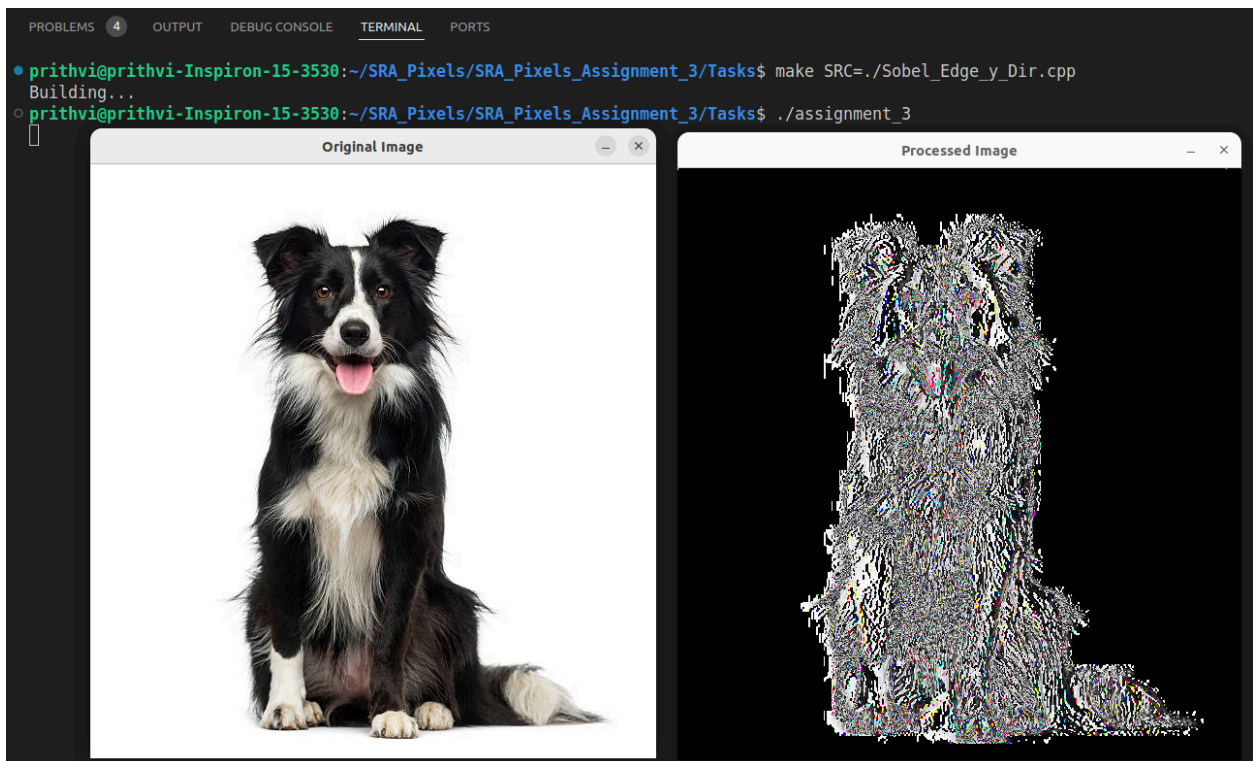
```
Sobel_Edge_y_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_y_Dir.cpp > ...
1  #include <iostream>
2  #include <opencv2/highgui.hpp>
3  #include <opencv2/imgcodecs.hpp>
4  #include <opencv2/imgproc.hpp>
5
6  using namespace std;
7  using namespace cv;
8
9  int main(){
10     Mat img;
11     img=imread("dog.jpg");
12     int img_wid=img.size().width;
13     int img_hgt=img.size().height;
14     Mat img1((img_hgt-2), (img_wid-2), CV_8UC3, Scalar(0, 0, 0));
15     imshow("Original Image", img);
16     for(int row=0; row<(img_hgt-2); row++){
17         for(int col=0; col<(img_wid-2); col++){
18             for(int i=0; i<3; i++){
19                 img1.at<Vec3b>(row, col)[i]=((-1)*(img.at<Vec3b>((row-1), (col-1))[i]))+((-2)*(img.at<Vec3b>(row, (col-1))[i]))+((-1)*(img.at<Vec3b>(row+1, (col-1))[i]));
20                 if(img1.at<Vec3b>(row, col)[i]>255){
21                     img1.at<Vec3b>(row, col)[i]=255;
22                 }
23             }
24         }
25     }
26     imshow("Processed Image", img1);
27     waitKey(0);
28     return 0;
29 }
```

```
Sobel_Edge_y_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_y_Dir.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 +((-1)*img.at<Vec3b>((row+1), (col-1))[i]))+(0*(img.at<Vec3b>((row-1), col)[i]))+(0*(img.at<Vec3b>(row, col)[i]))+(0*(img.at<Vec3b>((row+1), col)[i]))
20
21
22
23
24
25
26
27
28
29
```

```
Sobel_Edge_y_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_y_Dir.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 g.at<Vec3b>((row+1), col)[i]))+(1*(img.at<Vec3b>((row-1), (col+1))[i]))+(2*(img.at<Vec3b>(row, (col+1))[i]))+(1*(img.at<Vec3b>((row+1), col)[i]))
20
21
22
23
24
25
26
27
28
29
```

```
Sobel_Edge_y_Dir.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sobel_Edge_y_Dir.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 , col)[i]))+(1*(img.at<Vec3b>((row-1), (col+1))[i]))+(2*(img.at<Vec3b>(row, (col+1))[i]))+(1*(img.at<Vec3b>((row+1), (col+1))[i])));
20
21
22
23
24
25
26
27
28
29
```

Q.2 OUTPUT :



Q.3 CODE :

Guassien_Blur.cpp 4 X

SRA_Pixels_Assignment_3 > Tasks > Guassien_Blur.cpp > ...

```
1  #include <iostream>
2  #include <opencv2/highgui.hpp>
3  #include <opencv2/imgcodecs.hpp>
4  #include <opencv2/imgproc.hpp>
5
6  using namespace std;
7  using namespace cv;
8
9  int main(){
10     Mat img;
11     img=imread("waterfall.jpg");
12     int img_wid=img.size().width;
13     int img_hgt=img.size().height;
14     Mat img_gauss(img_hgt-2, (img_wid-2), CV_8UC3, Scalar(0, 0, 0));
15     imshow("Original Image", img);
16     int y=0;
17     {
18         loop1:for(int row=0; row<(img_hgt-2); row++){
19             for(int col=0; col<(img_wid-2); col++){
20                 for(int i=0; i<3; i++){
21                     img_gauss.at<Vec3b>(row, col)[i]=(((1*(img.at<Vec3b>((row-1), (col-1))[i]))+(2*(img.at<Vec3b>(row,
22                     if(img_gauss.at<Vec3b>(row, col)[i]>255){
23                         img_gauss.at<Vec3b>(row, col)[i]=255;
24                     }
25                 }
26             }
27         }
28     }
29     imshow("Processed Image", img_gauss);
30     waitKey(0);
31     return 0;
32 }
```

Guassien_Blur.cpp 4 X

SRA_Pixels_Assignment_3 > Tasks > Guassien_Blur.cpp > ...

```
9
10
11
12
13
14
15
16
17
18
19
20
21 ) [i]) + (1 * (img.at<Vec3b>((row+1), (col-1))[i])) + (2 * (img.at<Vec3b>((row-1), col)[i])) + (4 * (img.at<Vec3b>(row, col)[i])) + (
22
23
24
25
26
27
28
29
30
31
32
```

Guassian_Blur.cpp 4 X

SRA_Pixels_Assignment_3 > Tasks > Guassian_Blur.cpp > ...

```
9
10
11
12
13
14
15
16
17
18
19
20
21 c3b>((row+1), col)[i]))+(1*(img.at<Vec3b>((row-1), (col+1))[i]))+(2*(img.at<Vec3b>(row, (col+1))[i]))+(1*(img.at<Vec3b>
22
23
24
25
26
27
28
29
30
31
32
```

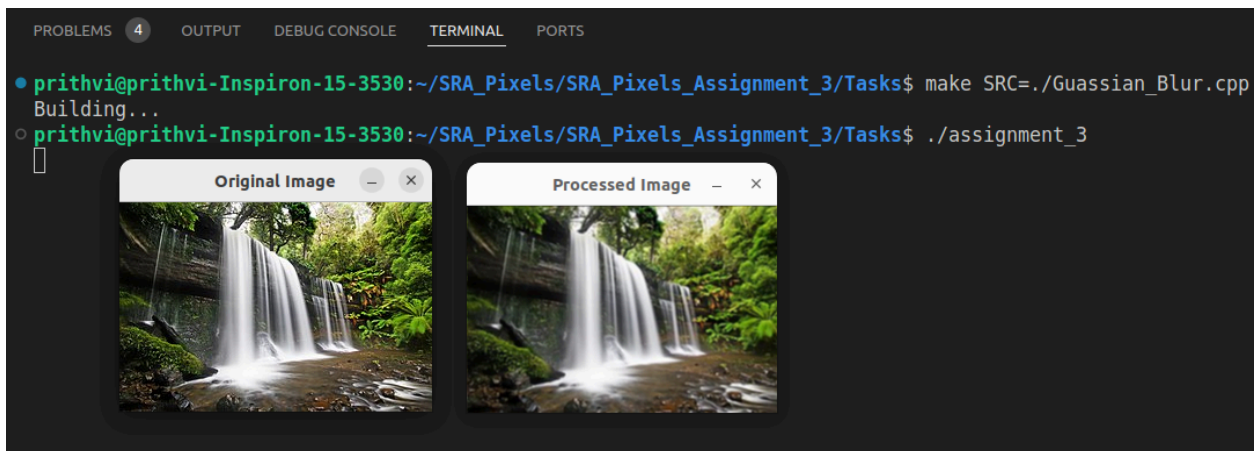
Guassian_Blur.cpp 4 X

SRA_Pixels_Assignment_3 > Tasks > Guassian_Blur.cpp > ...

```
9
10
11
12
13
14
15
16
17
18
19
20
21 ) [i]))+(1*(img.at<Vec3b>((row-1), (col+1))[i]))+(2*(img.at<Vec3b>(row, (col+1))[i]))+(1*(img.at<Vec3b>((row+1), (col+1)
22
23
24
25
26
27
28
29
30
31
32
```

```
Guassian_Blur.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Guassian_Blur.cpp > ...
9  int main(){
10     Mat img;
11     img=imread("waterfall.jpg");
12     int img_wid=img.size().width;
13     int img_hgt=img.size().height;
14     Mat img_gauss((img_hgt-2), (img_wid-2), CV_8UC3, Scalar(0, 0, 0));
15     imshow("Original Image", img);
16     int y=0;
17     {
18         loop1:for(int row=0; row<(img_hgt-2); row++){
19             for(int col=0; col<(img_wid-2); col++){
20                 for(int i=0; i<3; i++){
21                     img_gauss.at<Vec3b>(row, col)[i]=(((1*(img.at<Vec3b>((row-1), (col-1))[i]))+(2*(img.at<Vec3b>(row,
22                     if(img_gauss.at<Vec3b>(row, col)[i]>255){
23                         img_gauss.at<Vec3b>(row, col)[i]=255;
24                     }
25                 }
26             }
27         }
28     }
29     imshow("Processed Image", img_gauss);
30     waitKey(0);
31     return 0;
32 }
```

Q.3 OUTPUT :



Q.4 CODE :


```
Sharpen.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sharpen.cpp > ...
1 #include <iostream>
2 #include <opencv2/highgui.hpp>
3 #include <opencv2/imgcodecs.hpp>
4 #include <opencv2/imgproc.hpp>
5
6 using namespace std;
7 using namespace cv;
8
9 int main(){
10     Mat img;
11     img=imread("house.jpg");
12     int img_wid=img.size().width;
13     int img_hgt=img.size().height;
14     Mat img_sharp((img_hgt-2), (img_wid-2), CV_8UC3, Scalar(0, 0, 0));
15     imshow("Original Image", img);
16     int y=0;
17     {
18         loop1:for(int row=0; row<(img_hgt-2); row++){
19             for(int col=0; col<(img_wid-2); col++){
20                 for(int i=0; i<3; i++){
21                     img_sharp.at<Vec3b>(row, col)[i]=(0*(img.at<Vec3b>((row-1), (col-1))[i]))+((-0.5)*(img.at<Vec3b>(row, (col-1))[i]))
22                     if(img_sharp.at<Vec3b>(row, col)[i]>255){
23                         img_sharp.at<Vec3b>(row, col)[i]=255;
24                     }
25                 }
26             }
27         }
28     }
29     imshow("Processed Image", img_sharp);
30     waitKey(0);
31     return 0;
32 }
```

```
Sharpen.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sharpen.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 1))[i]))+0*(img.at<Vec3b>((row+1), (col-1))[i]))+((-0.5)*(img.at<Vec3b>((row-1), col)[i]))+(3*(img.at<Vec3b>(row, col)[i]))+((-0.5)*(
22
23
24
25
26
27
28
29
30
31
32
```

```
Sharpen.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sharpen.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 ))+((-0.5)*(img.at<Vec3b>((row+1), col)[i]))+(0*(img.at<Vec3b>((row-1), (col+1))[i]))+((-0.5)*(img.at<Vec3b>(row, (col+1))[i]))+(0*(img
22
23
24
25
26
27
28
29
30
31
32
```

```
Sharpen.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sharpen.cpp > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 )][i]))+(0*(img.at<Vec3b>((row-1), (col+1))[i]))+((-0.5)*(img.at<Vec3b>(row, (col+1))[i]))+(0*(img.at<Vec3b>((row+1), (col+1))[i])));
22
23
24
25
26
27
28
29
30
31
32
```

```
Sharpen.cpp 4 X
SRA_Pixels_Assignment_3 > Tasks > Sharpen.cpp > ...
9  int main(){
17  {
18      loop1:for(int row=0; row<(img_hgt-2); row++){
19          for(int col=0; col<(img_wid-2); col++){
20              for(int i=0; i<3; i++){
21                  img_sharp.at<Vec3b>(row, col)[i]=((0*(img.at<Vec3b>((row-1), (col-1))[i]))+((-0.5)*(img.at<Vec3b>(row, (col-1))[i]))
22                  if(img_sharp.at<Vec3b>(row, col)[i]>255){
23                      img_sharp.at<Vec3b>(row, col)[i]=255;
24                  }
25              }
26          }
27      }
28  }
29  imshow("Processed Image", img_sharp);
30  waitKey(0);
31  return 0;
32 }
```

Q.4 OUTPUT :

