

Computer Vision HW1 Report

NTU CSIE R08922024 Alfons Hwu

Part1

a

```
void upside_down()
{
    for (int i = 0; i < center_i; i++)
    {
        for (int j = 0; j < m; j++)
        {
            swap(img.at<uchar>(i, j), img.at<uchar>(n - i - 1, j));
        }
    }
}
```



Q

b

```
void rightside_left()
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < center_j; j++)
        {
            swap(img.at<uchar>(i, j), img.at<uchar>(i, m - j - 1));
        }
        cout << endl;
    }
}
```

Q

Q

Q



Q
Q

c

```
void diag_mirrored()  
{  
    upside_down();  
    rightside_left();  
}
```

Q

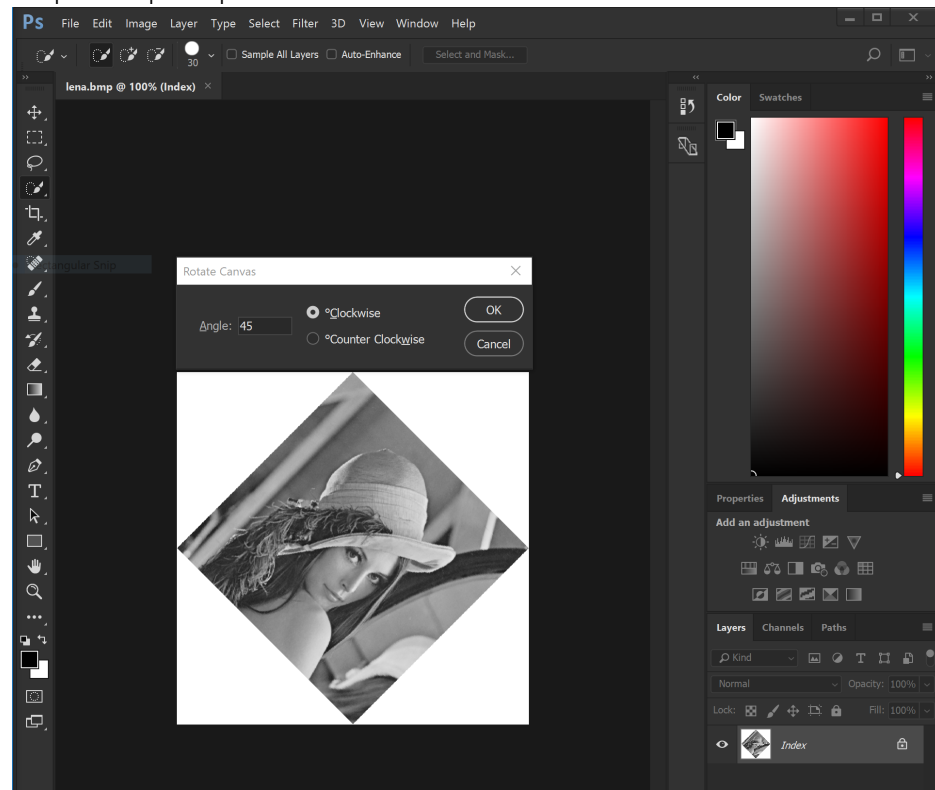


Q
Q
Q

Part2

d

Use photoshop to implement this



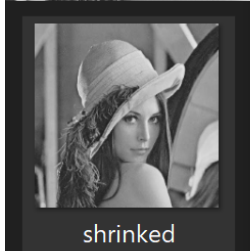
e

```

void shrink()
{
    Mat shrunked(n / 2, m / 2, CV_8UC1, Scalar(0, 0, 0));
    for (int i = 0; i < n; i += 2)
    {
        for (int j = 0; j < m; j += 2)
        {
            shrunked.at<uchar>(i / 2, j / 2) = img.at<uchar>(i, j);
        }
    }
    img = shrunked;
}

```

We can clearly see that the size of photo has been shirnked into half



shrunked Properties

General Security Details Previous Versions

Property	Value
Origin	
Date taken	
Image	
Dimensions	256 x 256
Width	256 pixels
Height	256 pixels
Bit depth	8
File	
Name	shrunked.png
Item type	PNG File
Folder path	I:\programming\CV_Fall_2019\HW1\hw_t...
Date created	9/22/2019 10:49 AM
Date modified	9/22/2019 10:49 AM
Size	45.8 KB

f

```

void binarize()
{
    unsigned char val = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            val = (unsigned char)img.at<uchar>(i, j);
            img.at<uchar>(i, j) = (val > (unsigned char) 0x7F) \
                ? 0xFF : 0x0;
        }
    }
}

```

If > 127m then set as white(0xFF), otherwise, set black (0x0)

