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));
		}
	}
}
```

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;
	}
}
```

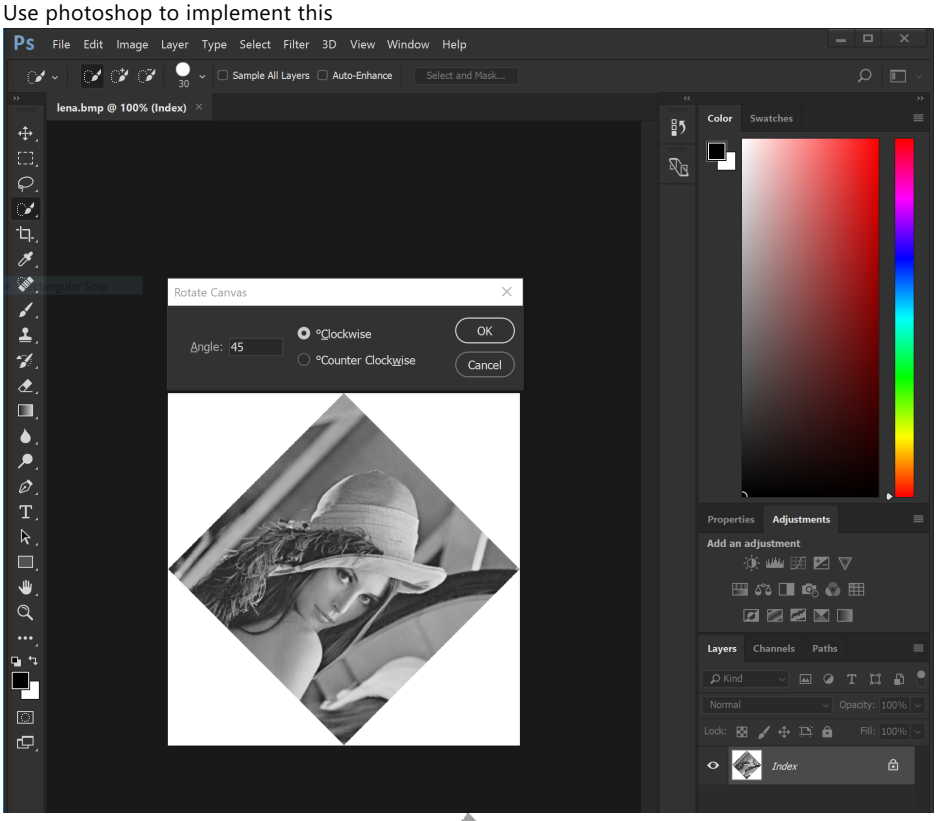
c

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

Part2

d

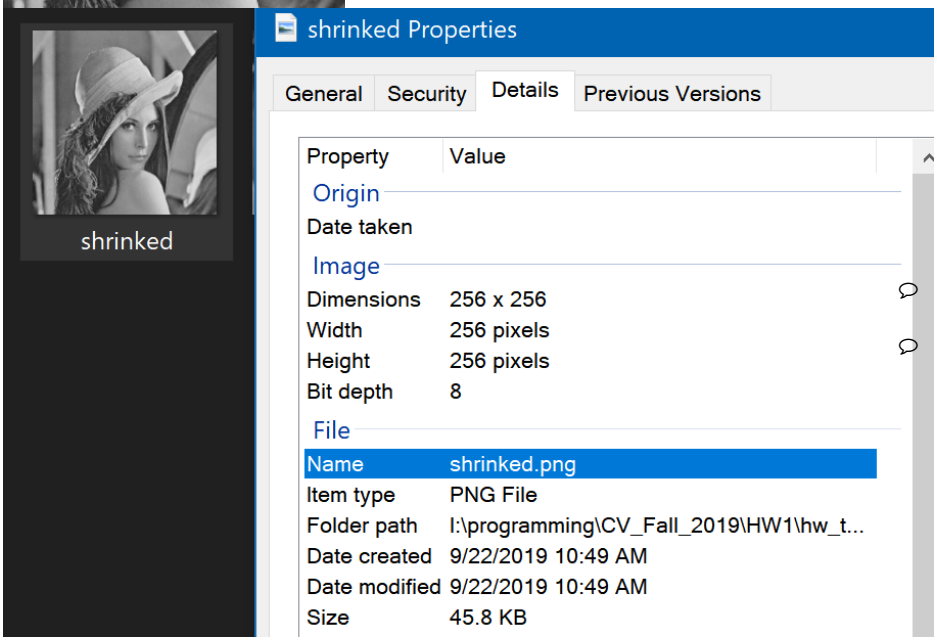
Use photoshop to implement this



e

```
void shrink()
{
	Mat shrinked(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)
		{
			shrinked.at<uchar>(i / 2, j / 2) = img.at<uchar>(i, j);
		}
	}
	img = shrinked;
}
```

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



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)