Assignment1 圖像旋轉

程式功能:將一張圖像的(a)整張圖像，(b)中心內切圓區域，旋轉一個角度(逆時針旋 轉 0 度至 359 度)：利用一個滑動條(trackbar)控制旋轉角度。

開發環境: Windows 10 + Visual Studio 2019 + OpenCV 4.5.4

使用語言:C++

在全域變數的宣告:

```
Mat img = imread("yzu.bmp"); //原影像
int angle = 0;               //旋轉整張圖片用的角度
int circleAngle = 0;         //旋轉圓形圖片的角度
Mat cirImage;                //裁切下來的圓形圖片
Mat mask = Mat::zeros(img.size(), CV_8UC1);    //圓形mask
```

(a) 旋轉整張圖像

方法:利用拉桿取得角度後，依角度旋轉。

用拉桿取得角度

使用 cv::createTrackbar(拉桿名稱, 放在哪個視窗, 值, 最大值, 回調函數, 傳給回調函數的值)

會呼叫回調函數 rotateImage，可在函數裡進行旋轉圖片。把值與會用到的圖以全域變數宣告可以減少傳來傳去的麻煩。

```
createTrackbar("Angle", "Image Rotation", &angle, 360, rotateImage);
```

圖(1) createTrackbar 函數程式碼

在回調函數中旋轉圖片

宣告一個 mat 型態的旋轉矩陣 rotated。

先找到選轉中心(圖高、寬的一半)，與拉桿取得的值得到一個選轉矩陣。

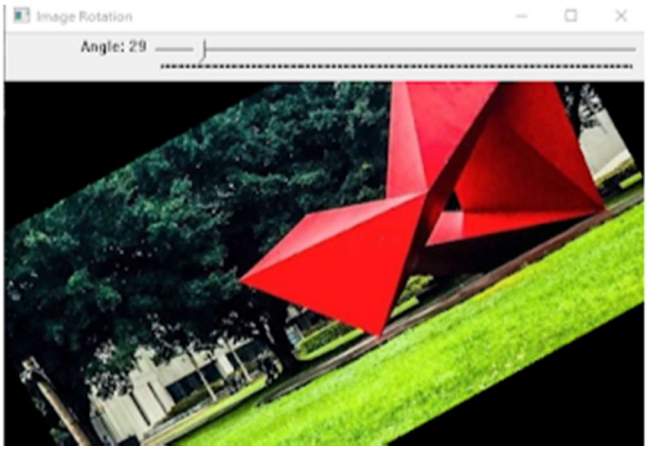
使用 cv::getRotationMatrix2D(選轉中心,角度,縮放)

再使用 cv::warpAffine(輸入圖像,輸出圖像,旋轉矩陣,輸出圖像的尺寸,插值方法(默認為線性插值),像素外推法（默認為 BORDER_CONSTANT）,border 值(默認為 0))

```
void rotateImage(int, void*) {
    Mat rotated;                //旋轉矩陣
    Point2f center(img.cols / 2.0, img.rows / 2.0);     //找圖中心
    Mat M = getRotationMatrix2D(center, angle, 1.0);    //計算出旋轉矩陣
    warpAffine(img, rotated, M, { img.size().width,img.size().height });    //旋轉

    imshow("Image Rotation", rotated);
}
```

圖(2) 回調函數 rotateImage 程式碼



圖(3) 成果 a

(b) 旋轉整張圖像

方法:切下內切圓 > 使用拉桿選轉內切圓(與(a)旋轉圖片方法相同) > 把旋轉後的圓形貼回原圖片上

裁切成圓形圖片

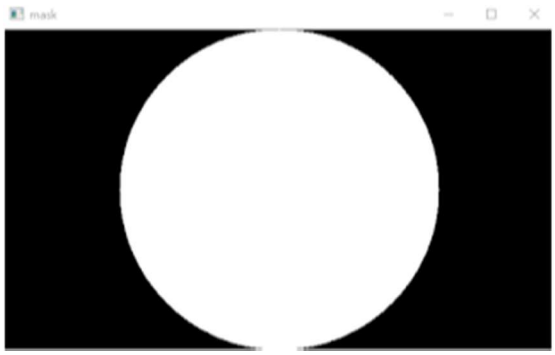
用圓形的 mask 遮罩留下中心內切圓區域:

先找出圓心(圖中心)，再找出半徑(圖 2 邊長較短邊)，使用

cv::circle(圖像,圓心坐標,圓形的半徑,線條的顏色,線條的粗細(正數>圓的線條的粗細程度，負數>填充),線條的類型,圓心坐標點和半徑值的小數點位數)

在大小跟原圖一樣的全黑圖上畫出白色圓形來做遮罩，再用

img1.copyTo(img2,mask)製作剪下內切圓，img2 得到的是 img1 被 mask 掩蓋後的圖，mask 中黑色是被遮蓋的地方。



圖(4)遮罩圓

圖(5) 內切圓圖

```
//製作圓形mask>圓形圖
Point center(img.cols / 2, img.rows / 2);    //圖心(圓心所在地)
int radius = min(img.cols, img.rows) / 2;    //半徑(圖的最小邊)
circle(mask, center, radius, Scalar(255), -1);  //在mask上畫出一個白色圓
img.copyTo(cirImage, mask);                  //印下圖片中間圓型，做成圓形圖

imshow("Circle Image Rotation", img);
```

圖(6)裁切內切圓形圖程式碼

使用拉桿選轉內切圓(與(a)旋轉圖片方法相同)

```
createTrackbar("Angle", "Circle Image Rotation", &circleAngle, 360, rotateCirImage);
```

圖(7)拉桿程式碼

```
void rotateCirImage(int, void*) {

    Mat rotatedcir;             //旋轉矩陣
    Point2f circenter(cirImage.cols / 2.0, cirImage.rows / 2.0);        //找圖中心
    Mat cirM = getRotationMatrix2D(circenter, circleAngle, 1.0);    //計算出旋轉矩陣
    warpAffine(cirImage, rotatedcir, cirM, { cirImage.size().width,cirImage.size().height });//旋轉
```

圖(8)旋轉內切圓形圖程式碼

把旋轉後的圓形貼回原圖片上

使用 img1.copyTo(img2,mask)，img2 是原圖的複製(img.clone())，img1 是上一步驟得到的旋轉內切圓，mask 是上上步驟使用到的圓形 mask。

```
Mat rotatecirimg = img.clone();     //複製一份原圖(b)用
rotatedcir.copyTo(rotatecirimg, mask);  //把選轉後的內切原圖切下中心圓形部分(有圖的部分)貼放到複製圖中間
```

圖(9) 旋轉後的圓形貼回原圖片程式碼