Rotation

Same like Translation, Rotation can be done using two ways.

1) Using imutils Packge
2) Using warpAffine function.

To Rotate, basically we neede to Compute the Center of the image.

And to Compute the Center of the image, we already seen that width and height //2

(h, w) = image.Shape[:2]

(Center x, center y) = (w//2, h//2)

Now will see how do it using warpAffine. For that we will be using a function called

getRotationMatrix2D()

M = cv2.getRotationMatrix2D((Centerx, center y), 45, 1.0) if the Second parameter is positive → It rotates the image in Anticlockwise direction.

if the Second Parameter is Negative → It rotates the image in Clockwise Direction.

Second Parameter Basically indicates Degrees

Third Parameter basically indicate scale of the image

(ie) Scale (or) Size of the Image

Higher the Scale → Higher the Size of image
Lower the Scale → Lower the size of image.

```
rotated = cv2.warpAffine(image, M, (w,h))
cv2.imshow("Rotating by 45 degrees", rotated)
```

We have seen till now is to rotate the image by computing the Center of the image.

What if we want to rotate the image into an arbitrary point ? Rather than Center ?

We Can simply plugin the Values for (Center x , Center Y)

```
M = CV2. getRotationMatrix2D((10,10), 45, 1.0)

rotated = CV2.warpAffine(image, M, (w, h))

CV2.imshow("Rotated by Arbitrary Point", rotated)
```

2) Now we will use the imutils function. Degrees.

```
rotated = imutils.rotate(image, 180)

CV2.imshow("Rotated by 180 degrees", rotated)
```

Note → when we use imutils. rotate, what happens is that there is a possiblity of scenario where image can get cuttoffs while rotating the image.

So, to overcome such kind of scenario there is one more function.

```
imutils.rotate_bound()
```

This will not cut off the image at any angle, when we rotate the image

```
rotate_new = imutils.rotate_bound(image, -33)
cv2.imshow("Rotated without Cropping", rotated_new)
cv2.waitKey(0).
```