image transformation

1. **Translation**
2. **Rotation**
3. **Resizing**
4. **flipping**
5. **cropping**

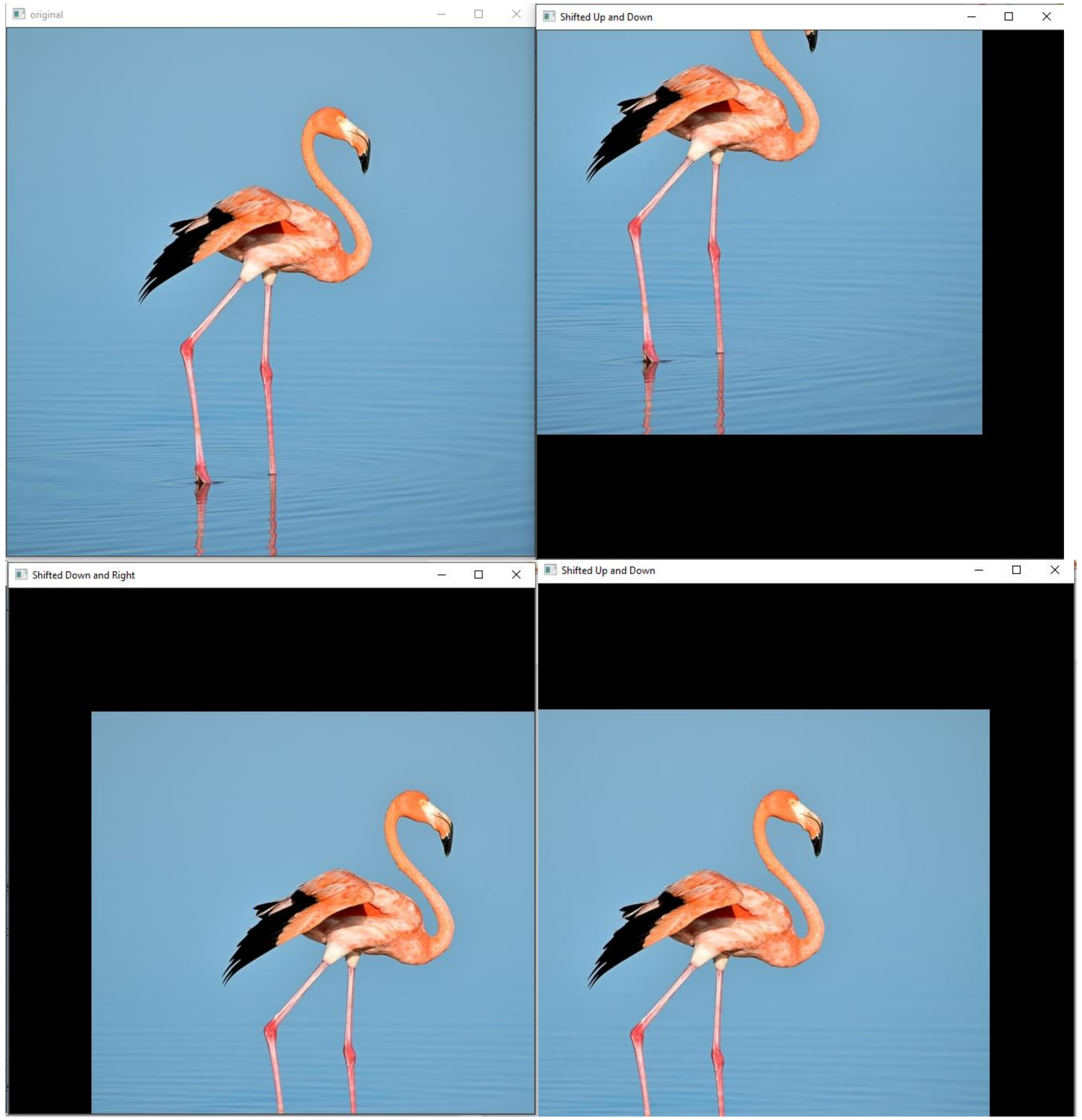
Translation

Image transformation

translation

- Translation is the shifting of an image along the x and y axis.
- Using translation, we can shift an image up, down, left, or right,

```
cv2.warpAffine(img,M,dsize)
```



translation matrix M

$$M = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{pmatrix}$$

$\boldsymbol{t_x}$: number of pixels will shift the image to right(+ve values) or left (-ve values)

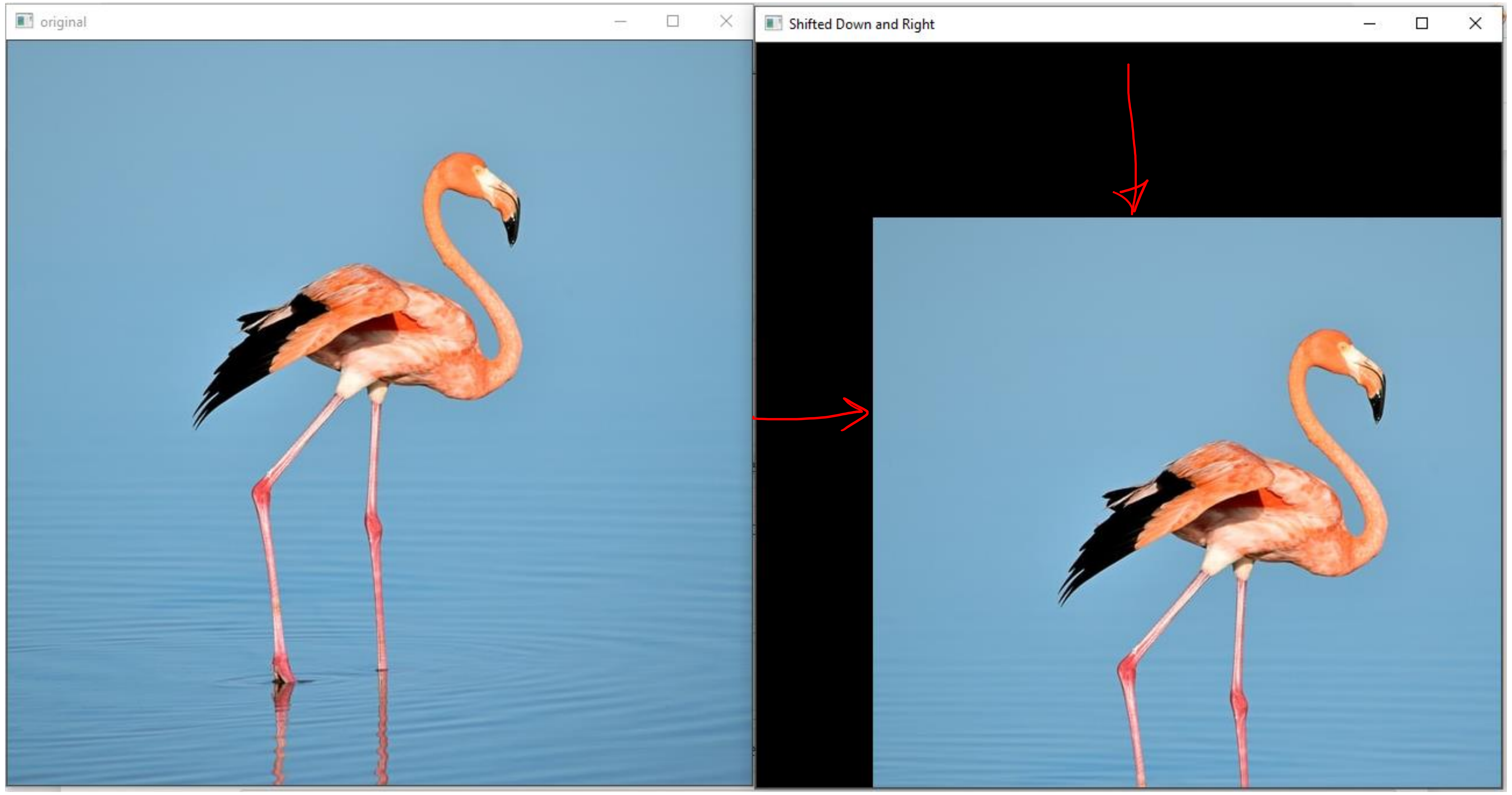
$\boldsymbol{t_y}$: number of pixels will shift the image to down (+ve values) or up (-ve values)

example

$$M = \begin{pmatrix} 1 & 0 & \underline{100} \\ 0 & 1 & \underline{150} \end{pmatrix}$$

t_x: image will shift 100 pixels right side

t_y: image will shift 150 pixels to down

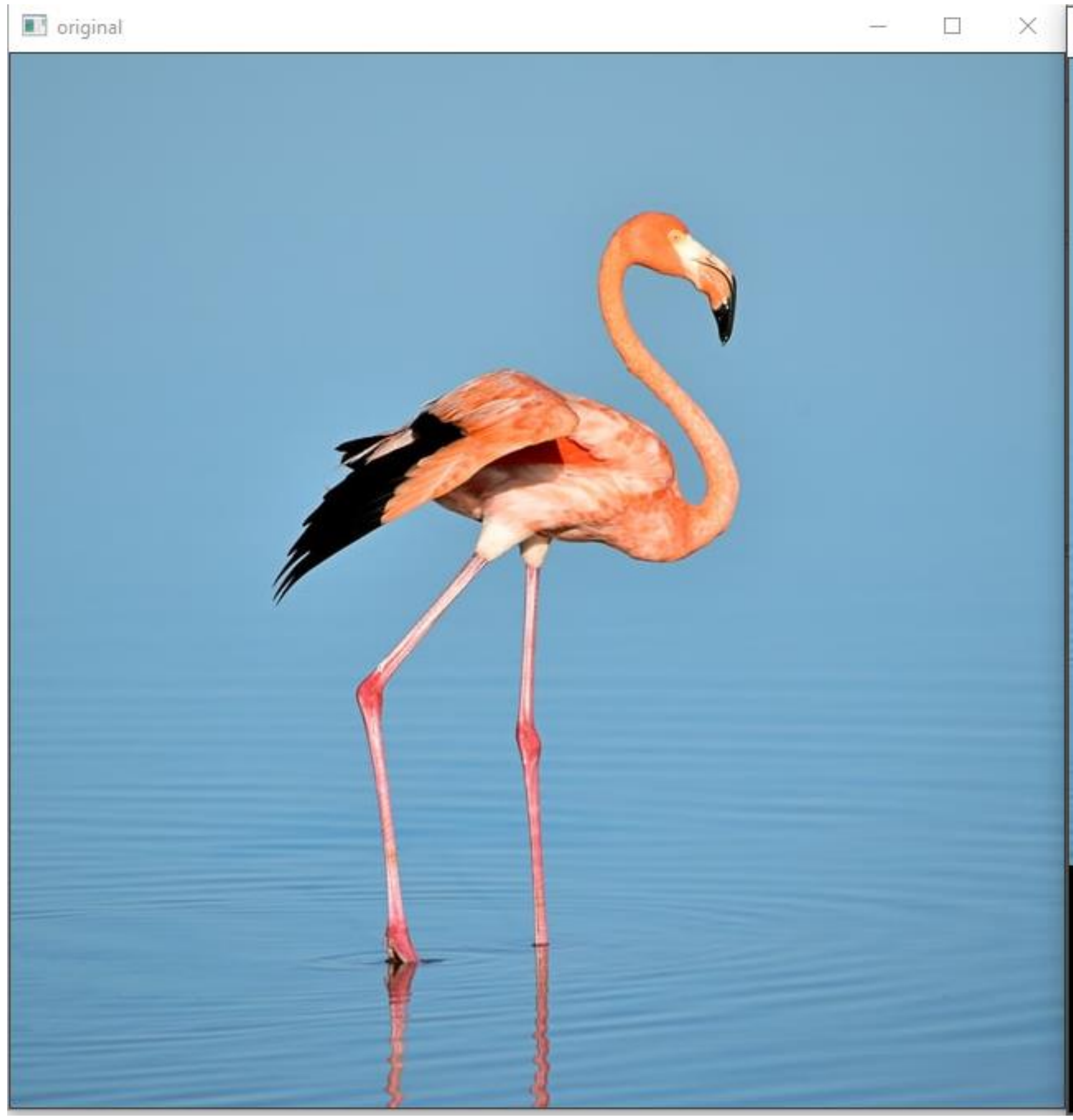


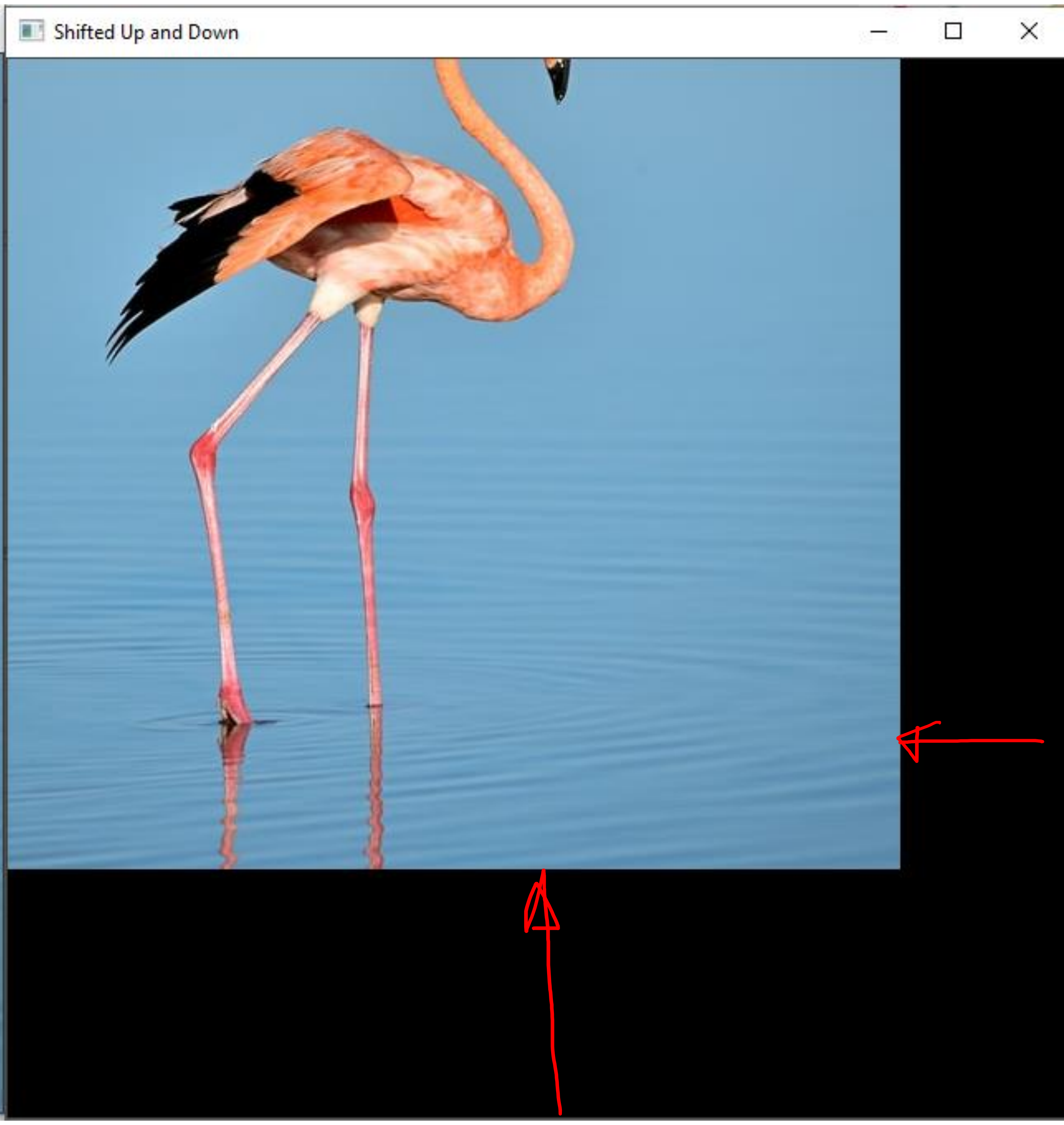
example

$$M = \begin{pmatrix} 1 & 0 & -100 \\ 0 & 1 & -150 \end{pmatrix}$$

$\boldsymbol{t_x}$: image will shift 100 pixels left side

$\boldsymbol{t_y}$: image will shift 150 pixels to upwards





Rotation

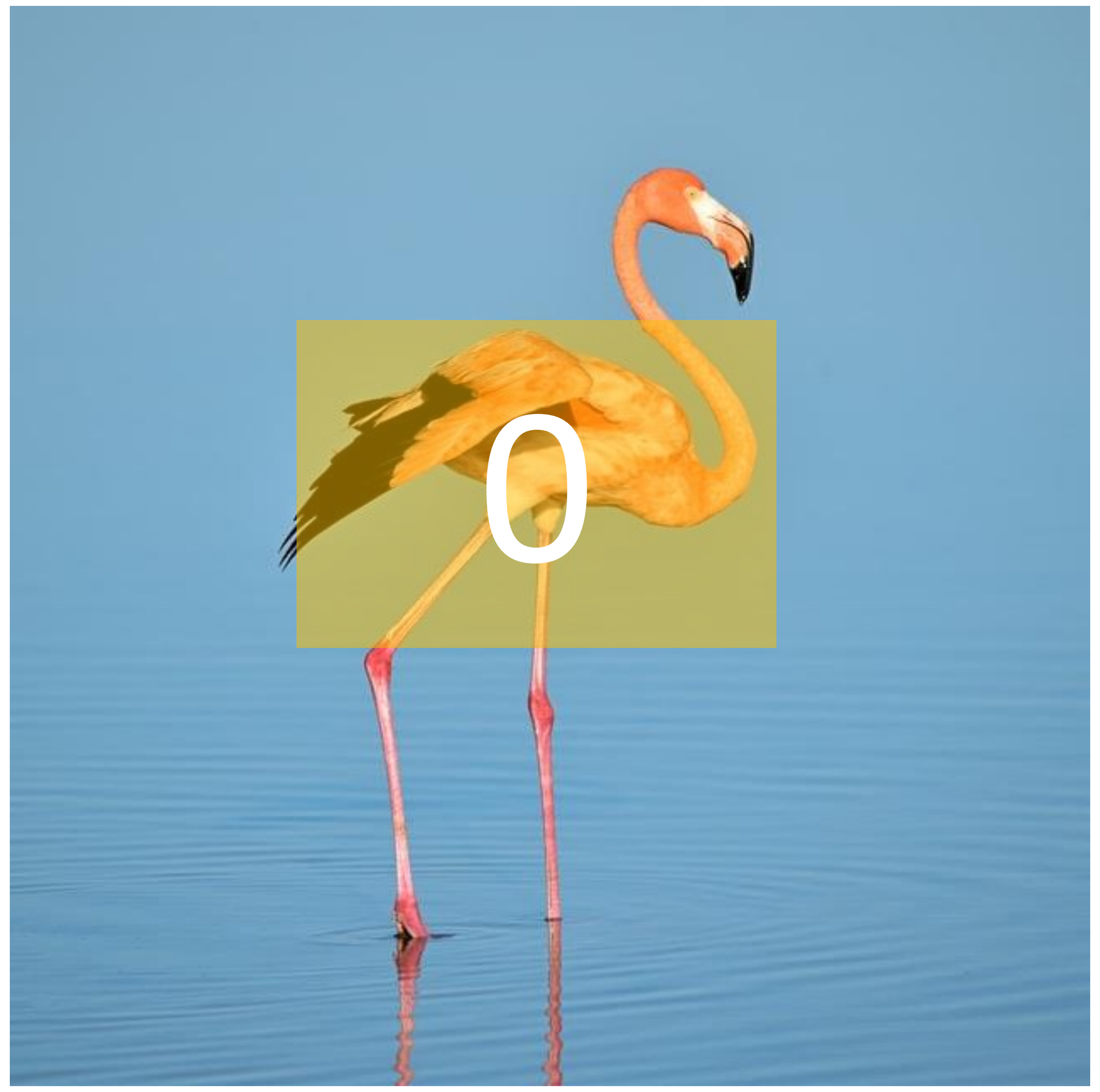
Image transformation

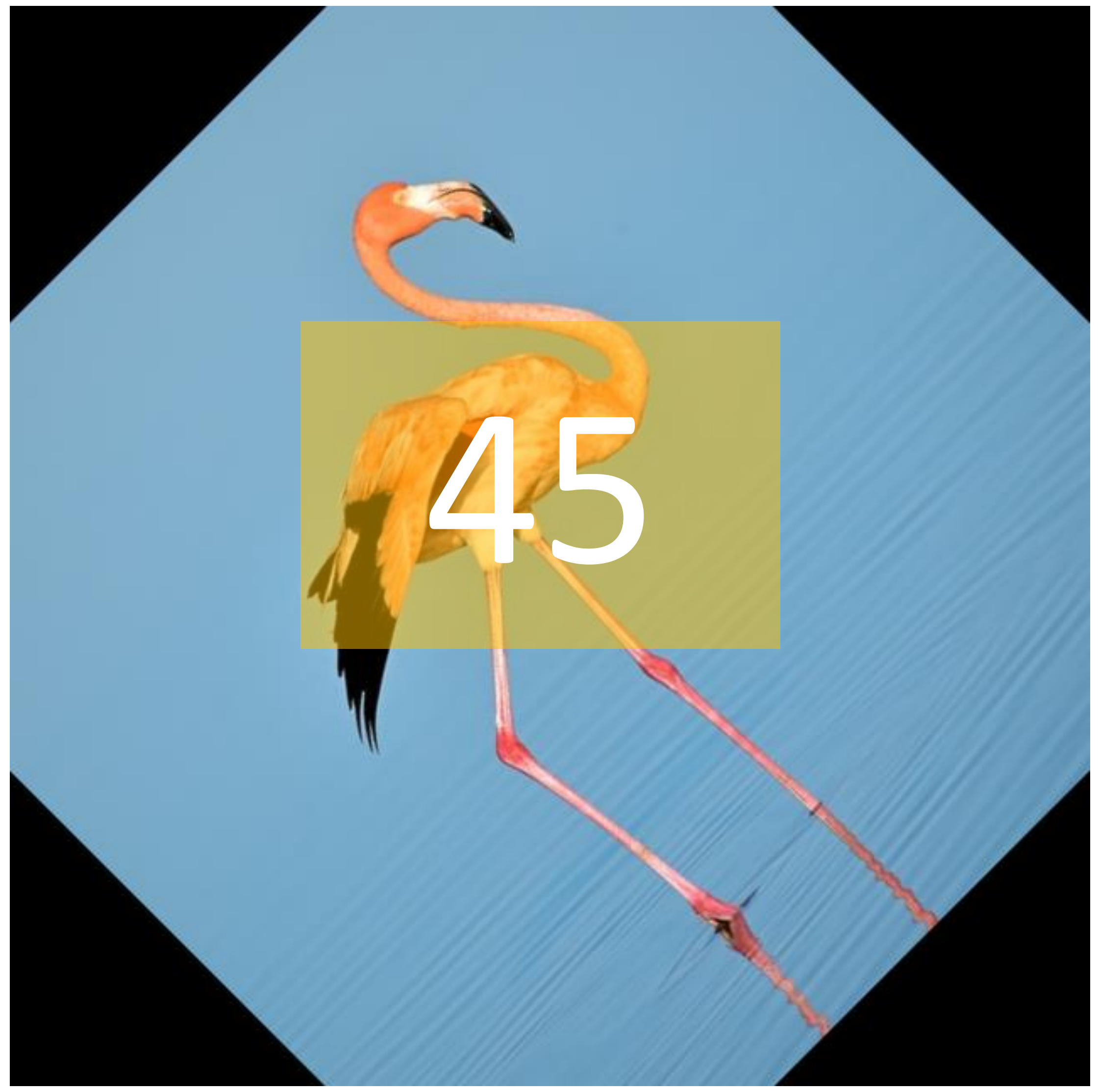
rotation

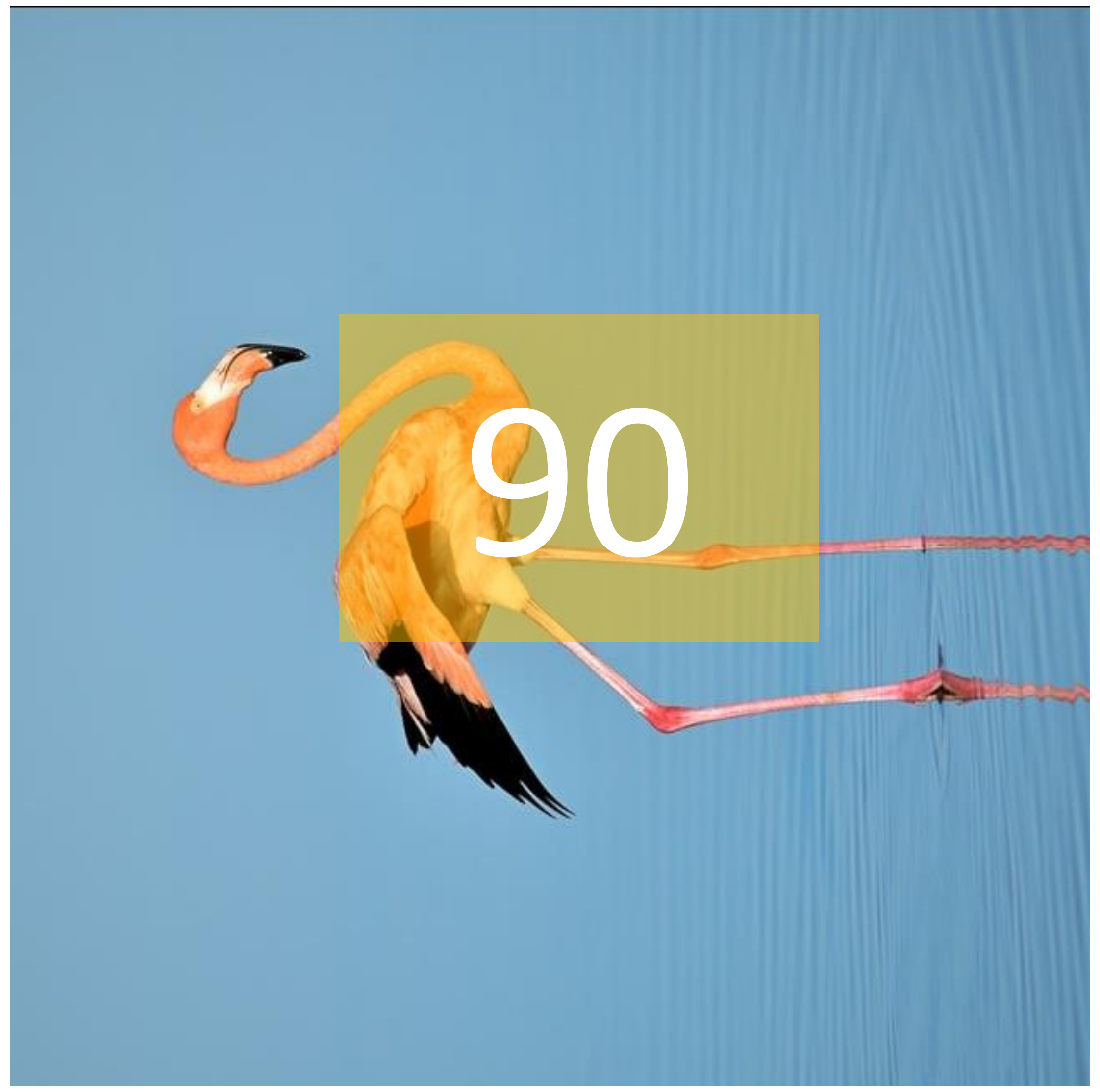
- We rotate an image for a given specific point.
- Use this we can rotate the image at any degree.

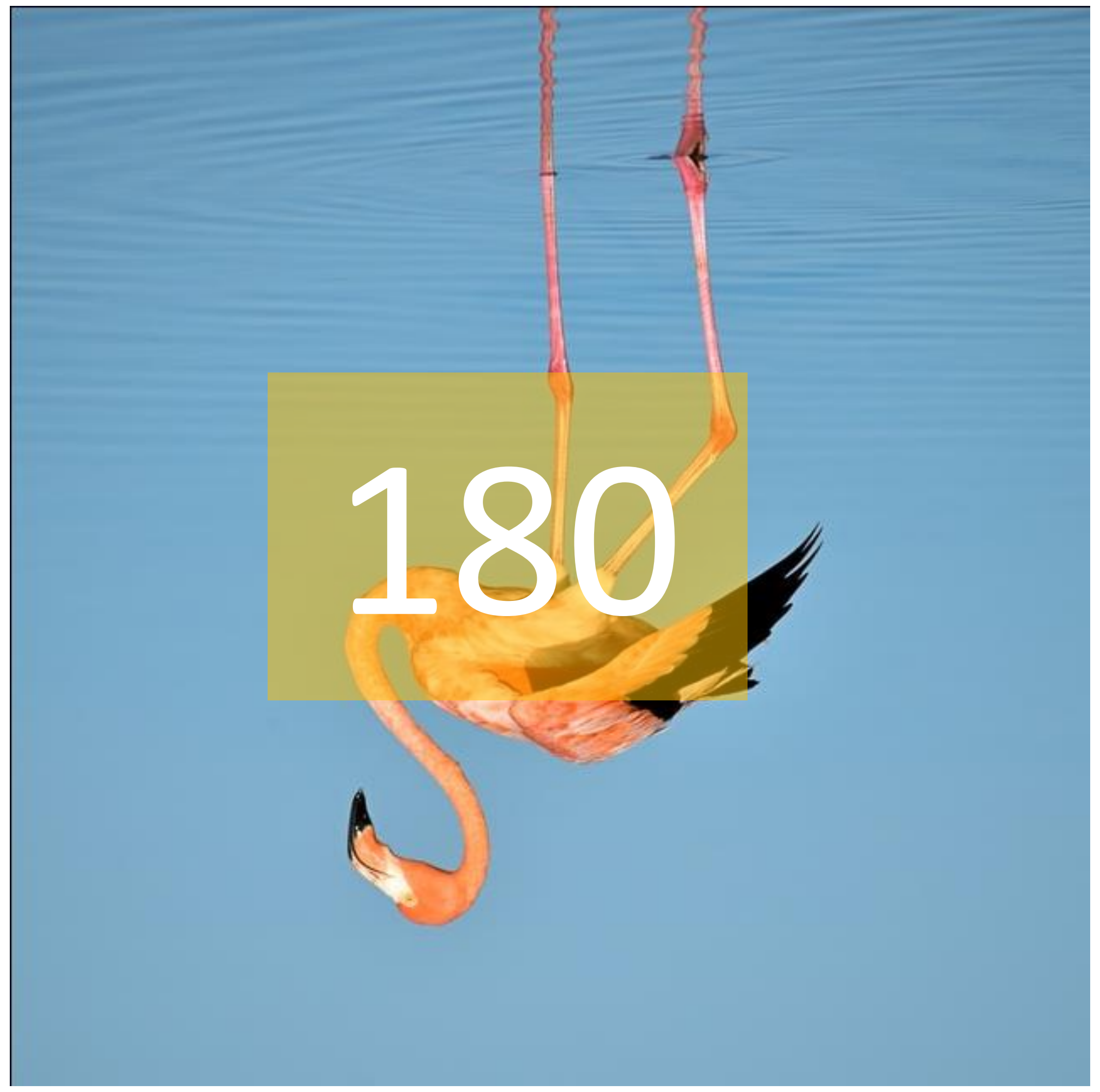
```
cv2.warpAffine(img,M,dsize)
```

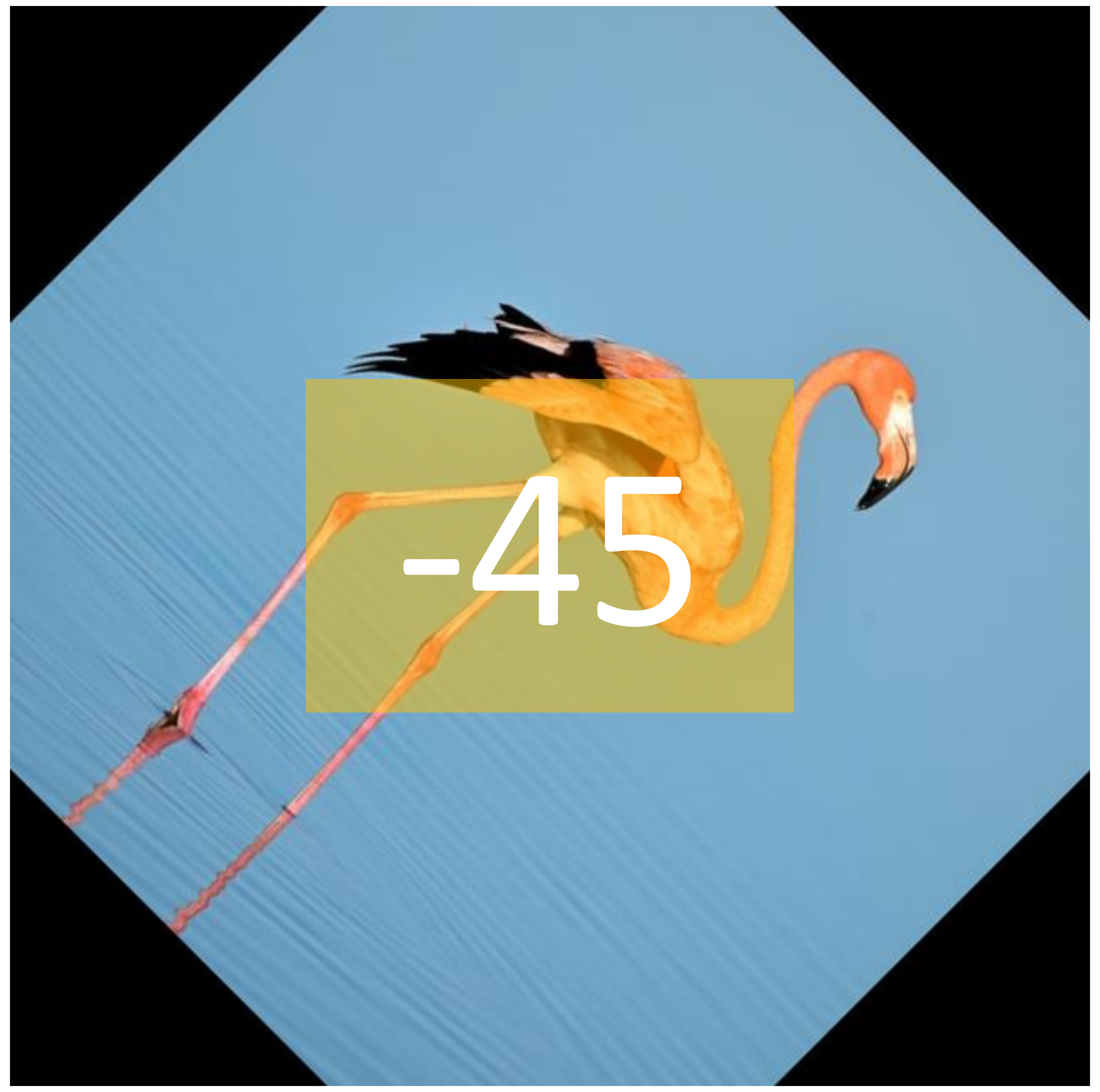
rotation

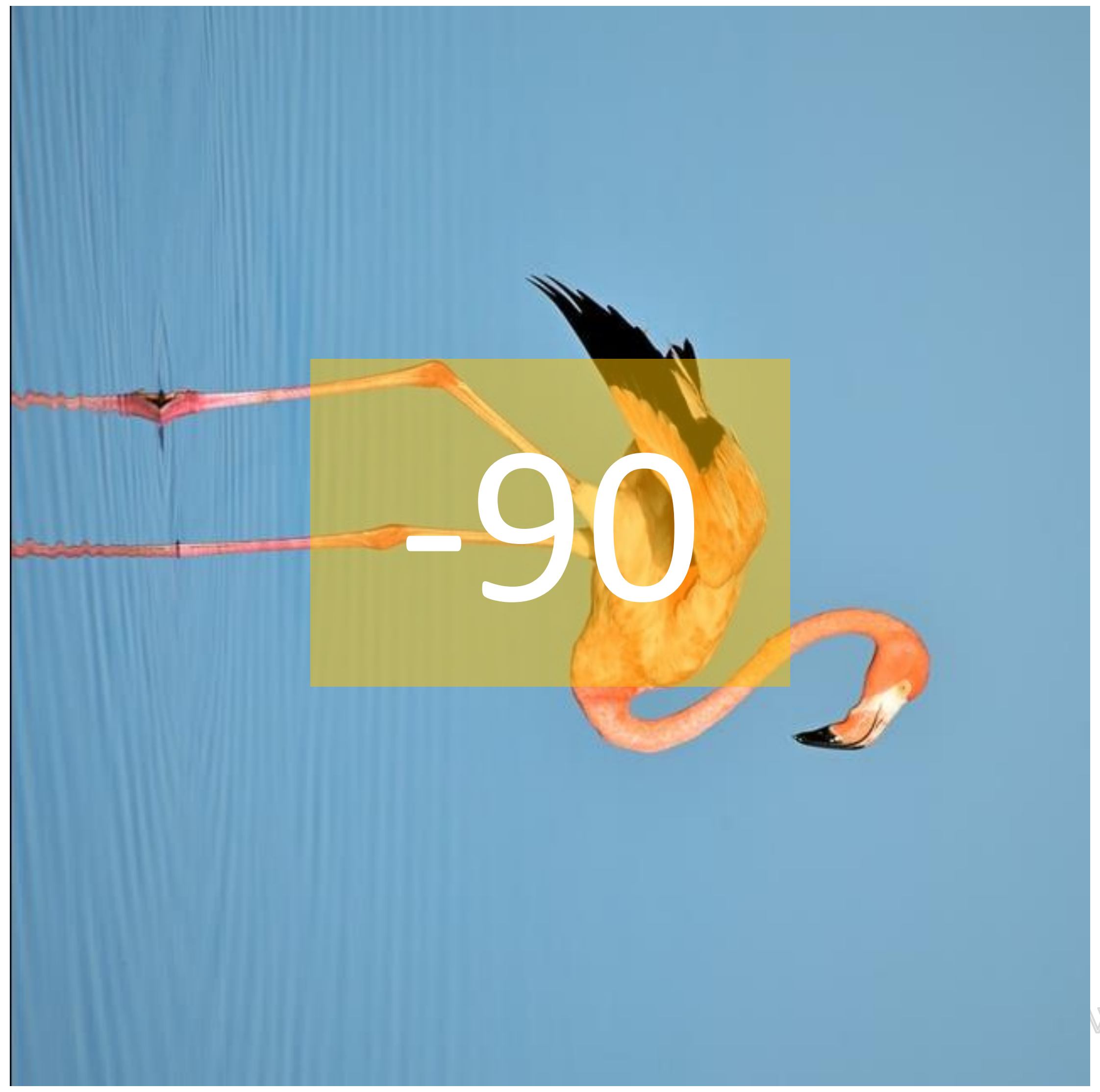












translation matrix M

$$M = \begin{pmatrix} \alpha & \beta & (1-\alpha) * c_x \; - \beta * c_y \\ -\beta & \alpha & \beta * c_x + (1-\alpha) * c_y \end{pmatrix}$$

$\boldsymbol{\alpha} = \mathbf{scale} \; * \mathbf{cos}(\boldsymbol{\theta})$

$\boldsymbol{\beta} = \boldsymbol{scale} \; * \boldsymbol{sin}(\boldsymbol{\theta})$

c_x, c_y is centre of image

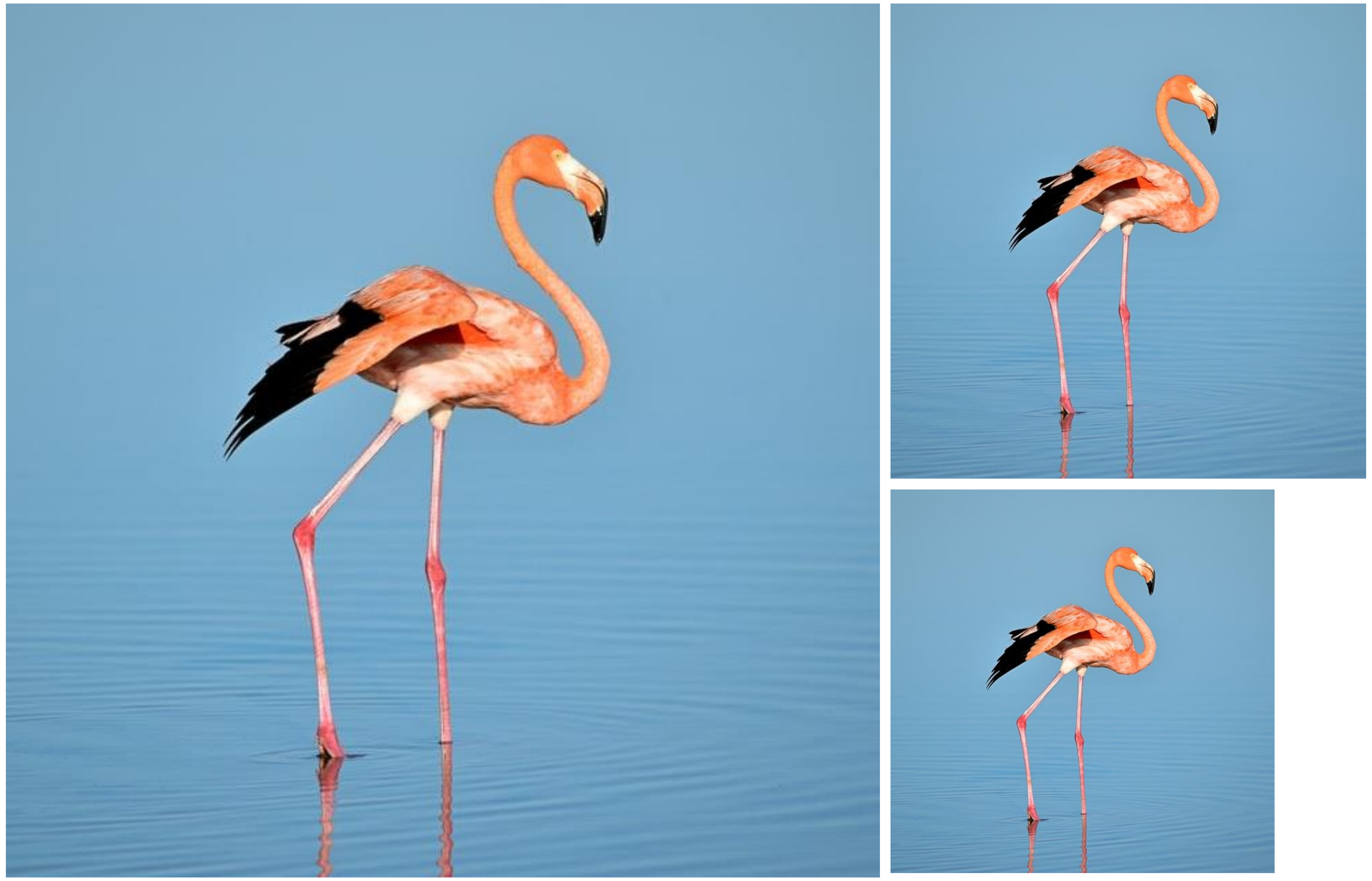
cv2.getRotationMatrix2D()

Resizing

Image transformation

resize

```
cv2.resize(img,dsize)
```

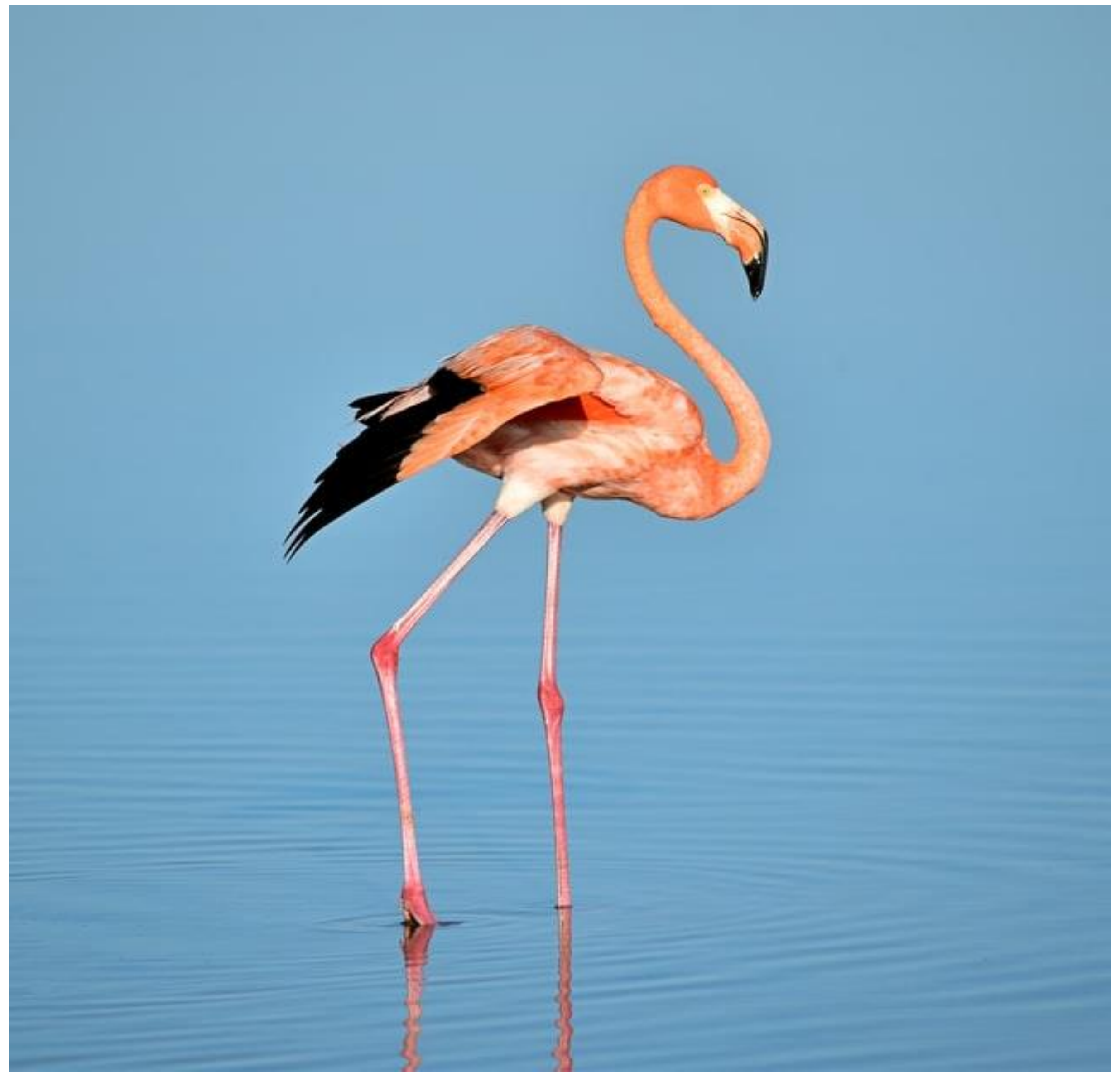


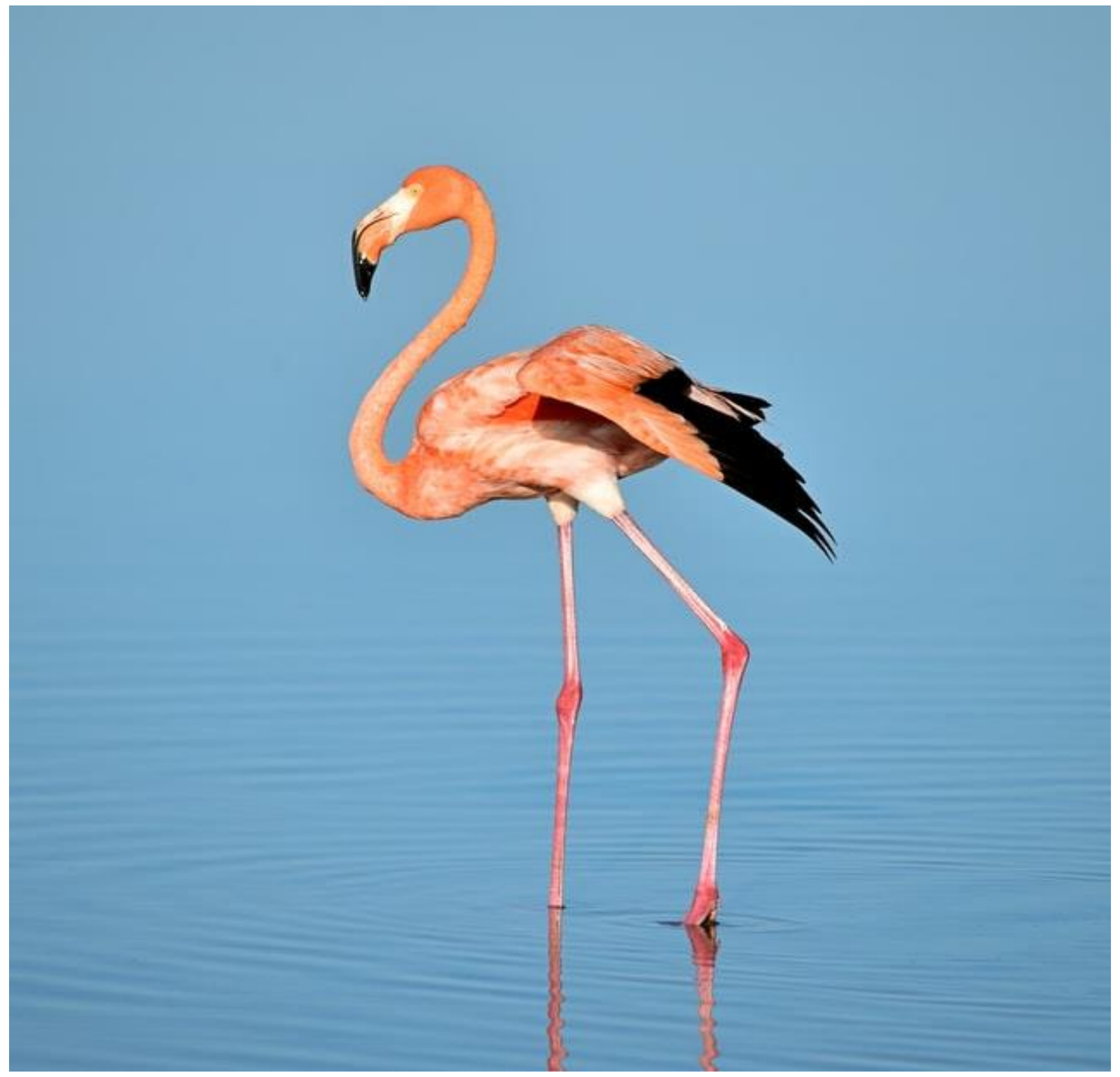
flipping

Image transformation

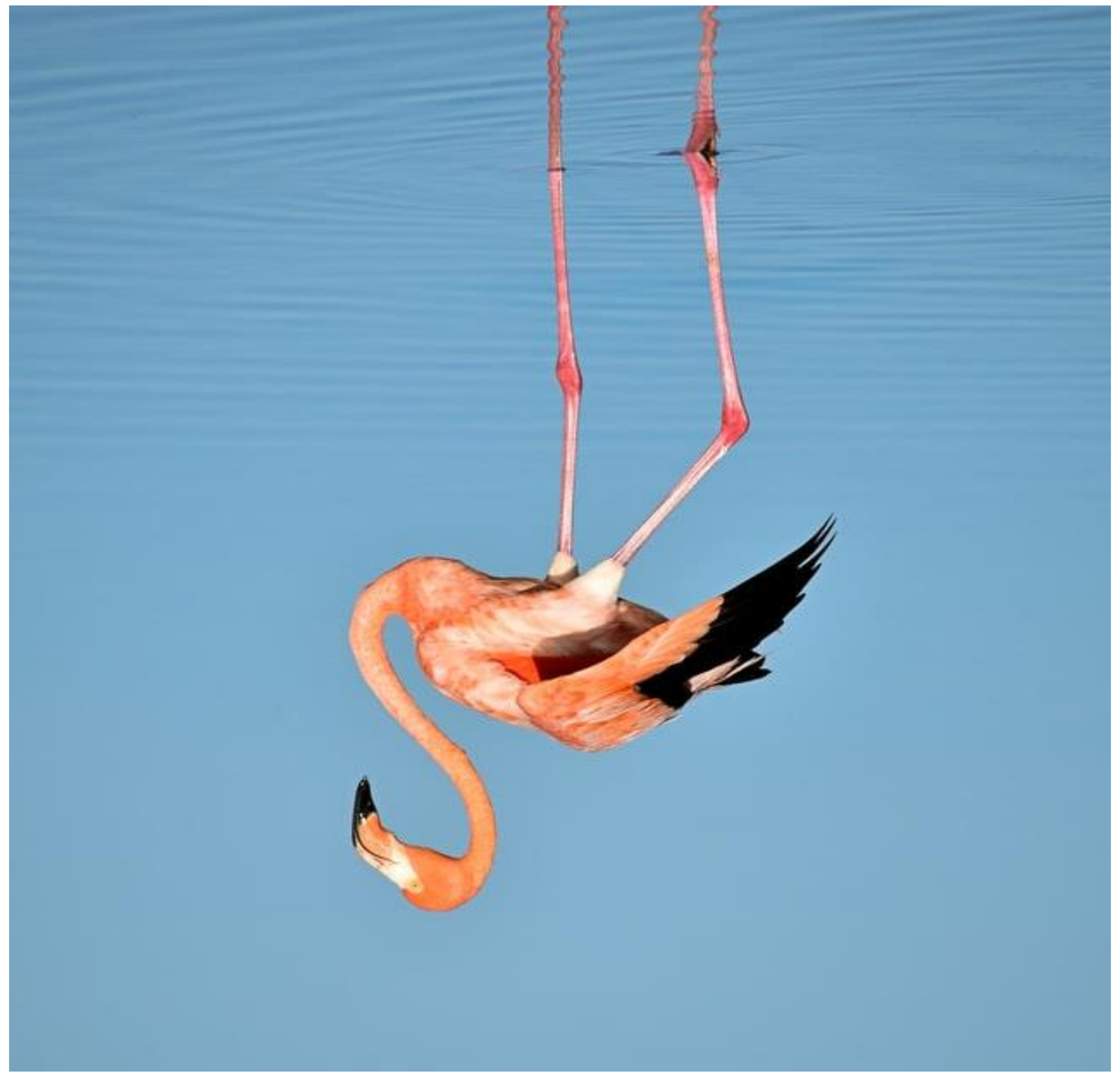
flipping

cv2.flip(img,flipCode)

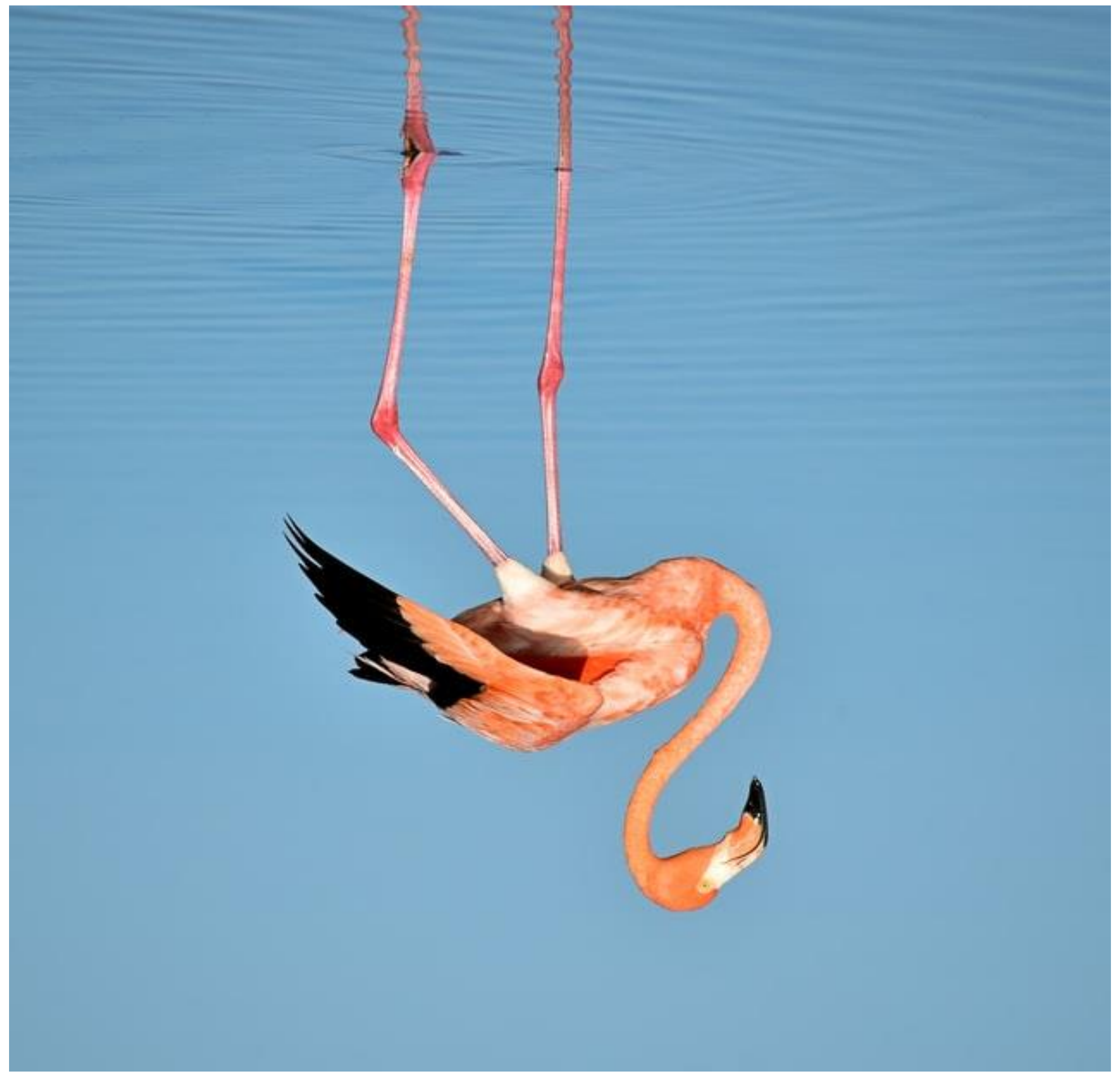




1



-1



0

next ▶

convert color in opencv python