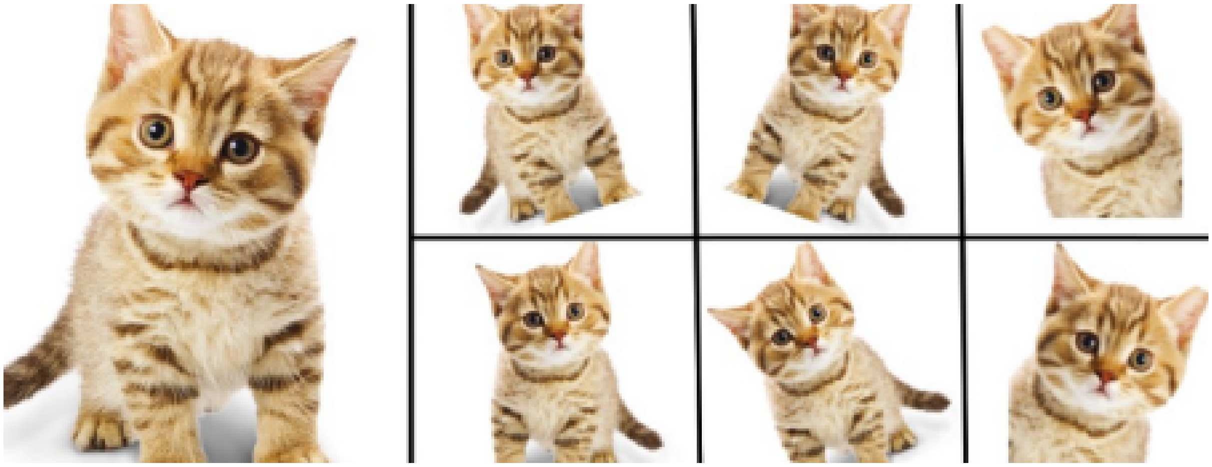
Image Processing for Computer Vision
Session 4

Geometric Transformation

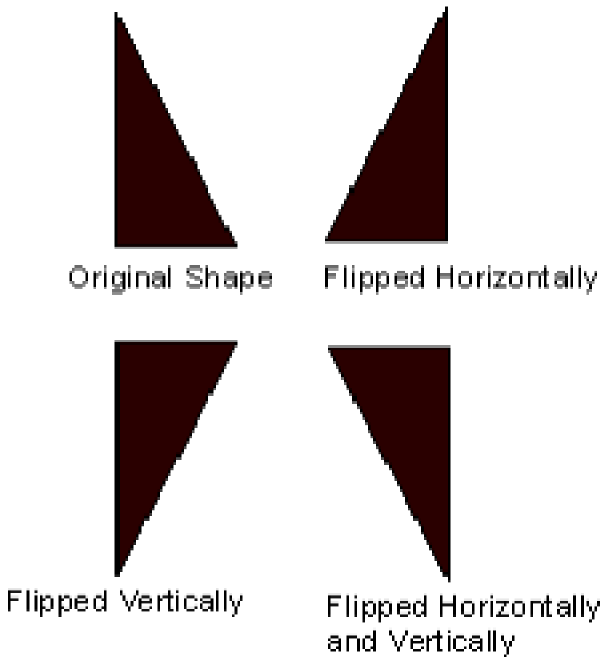


collected

Topics

- Flipping
- Cropping
- Resizing & Rescaling
- Rotation

Image Flipping


Original Shape
Flipped Horizontally
Flipped Vertically
Flipped Horizontally and Vertically

Flipping reverses the image along a specified axis (horizontal or vertical).

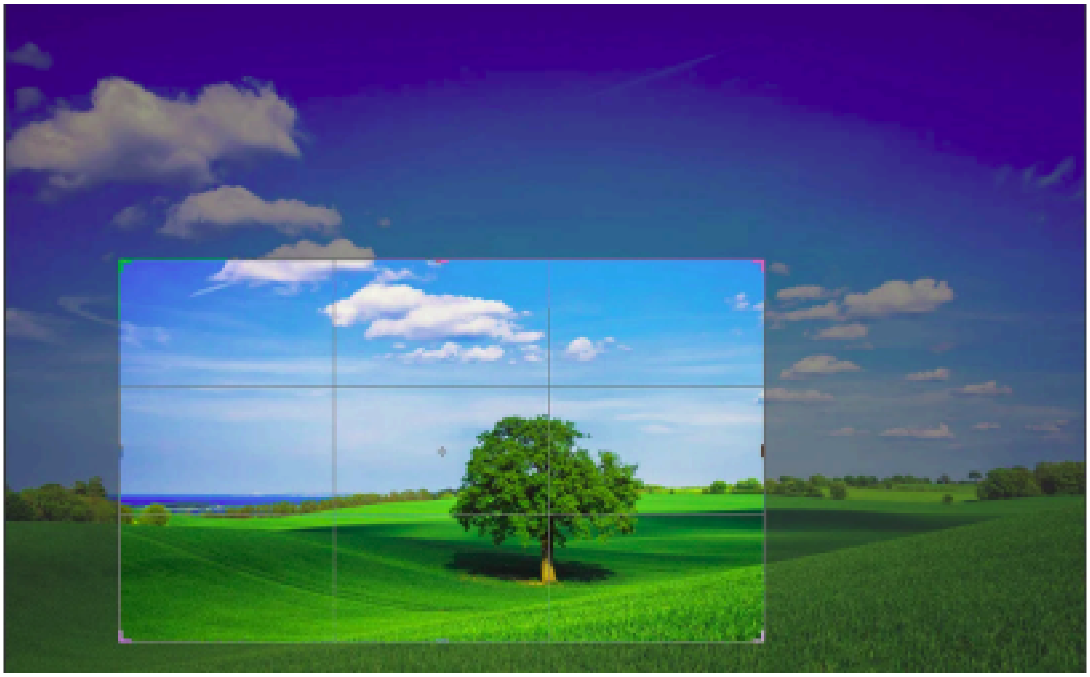
Syntax:

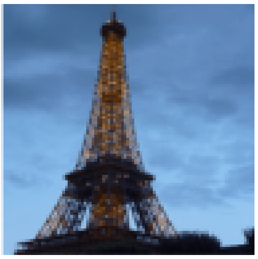
cv2.flip(src, flipCode)

src - loaded image array

flipCode - flipCode=0 for vertical flipping (around x-axis),
flipCode>0 for horizontal flipping (around y-axis),
flipCode<0 for flipping around both axis

Image Cropping

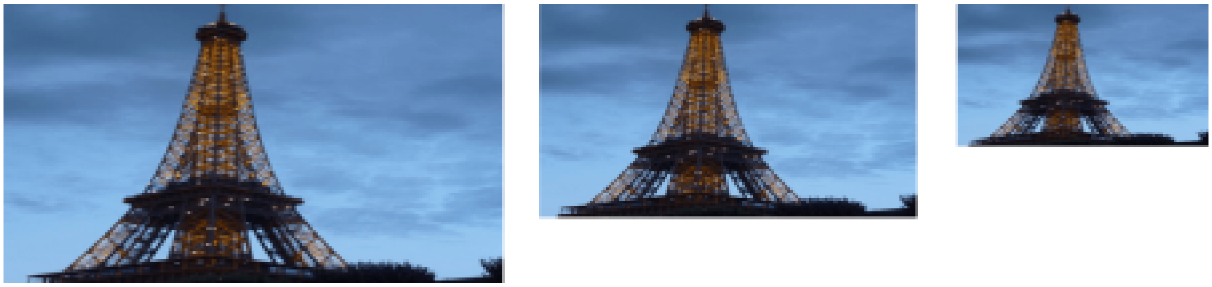




Removes unwanted area from the image. It's like slicing an array, when loaded as a numpy array.

original_image[y_start:y_end, x_start:x_end]

Rescaling & Resizing



Rescaling adjusts the size of an image by a scaling factor, maintaining the aspect ratio.
Resizing changes the dimensions of an image to a specified width and height.

cv2.resize(input_img, output_size, dest, fx, fy, interpolation)

Image Rotation

Images can be rotated using OpenCV by following two methods

Method-1:

cv2.rotate(src, code)
cv2. rotate() method is used to rotate a 2D array in multiples of 90 degrees.

Learn more about the rotation codes here

Method-2:

Step-1: making a rotation matrix
cv2.getRotationMatrix2D(center, angle, scale)
Calculates an affine matrix of 2D rotation. Creates a rotation matrix that will be used to rotate the image.

Step-2: apply the rotation rotation matrix and rotate the image
cv2.warpAffine(image, M, (w, h))
Here M = rotation matrix, (w,h) = The size of the output image
This applies the rotation matrix to the image to produce the rotated image.