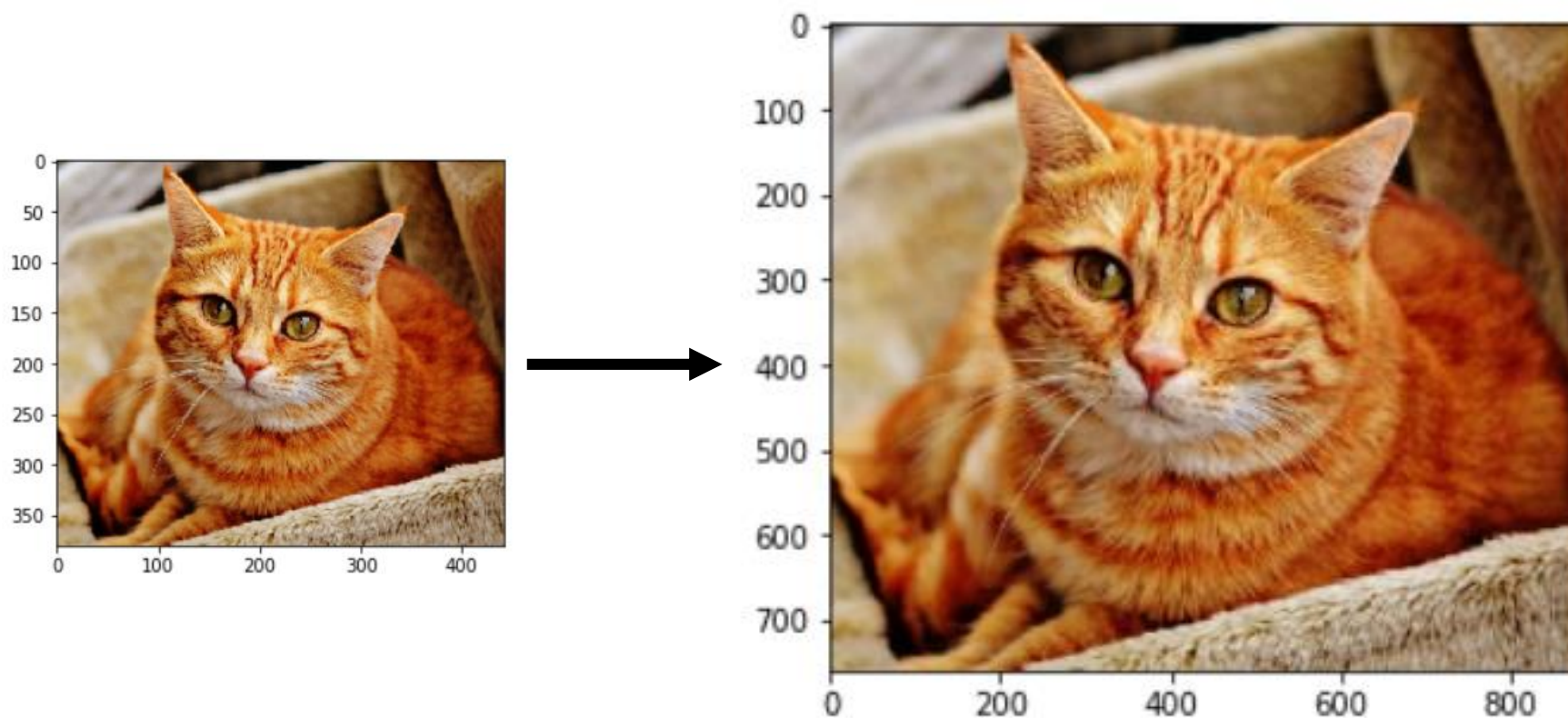


Python 데이터 분석과 이미지 처리

나동빈

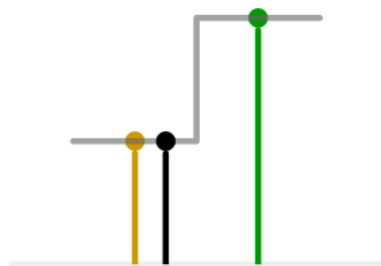
OpenCV 이미지 변형

이미지 크기 조절

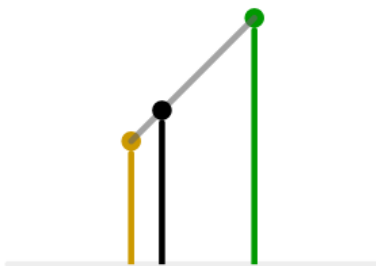


OpenCV 이미지 변형

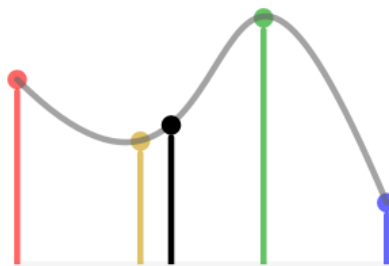
보간법(Interpolation)



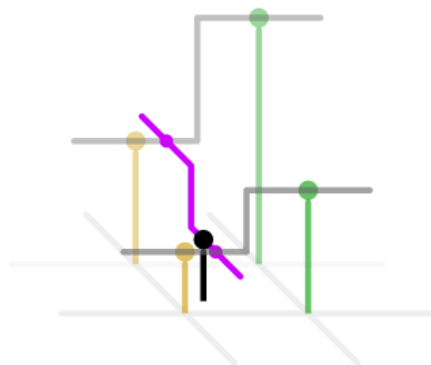
1D nearest-neighbour



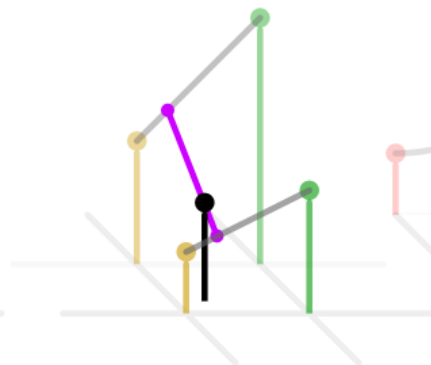
Linear



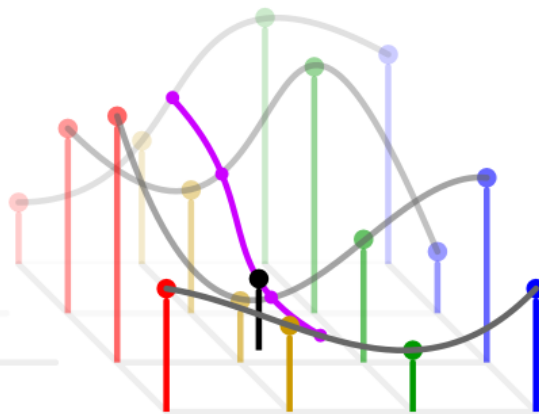
Cubic



2D nearest-neighbour



Bilinear



Bicubic

OpenCV 이미지 변형

이미지 크기 조절

<code>cv2.resize(image, dsize, fx, fy, interpolation)</code>	이미지의 크기를 조절합니다.
--	-----------------

- dsize: Manual Size

- fx: 가로 비율

- fy: 세로 비율

- interpolation: 보간법

INTER_CUBIC: 사이즈를 크게 할 때 주로 사용합니다.

INTER_AREA: 사이즈를 작게 할 때 주로 사용합니다.

보간법은 사이즈가 변할 때 픽셀 사이의 값을 조절하는 방법을 의미합니다.

OpenCV 이미지 변형

이미지 크기 조절

```
import cv2

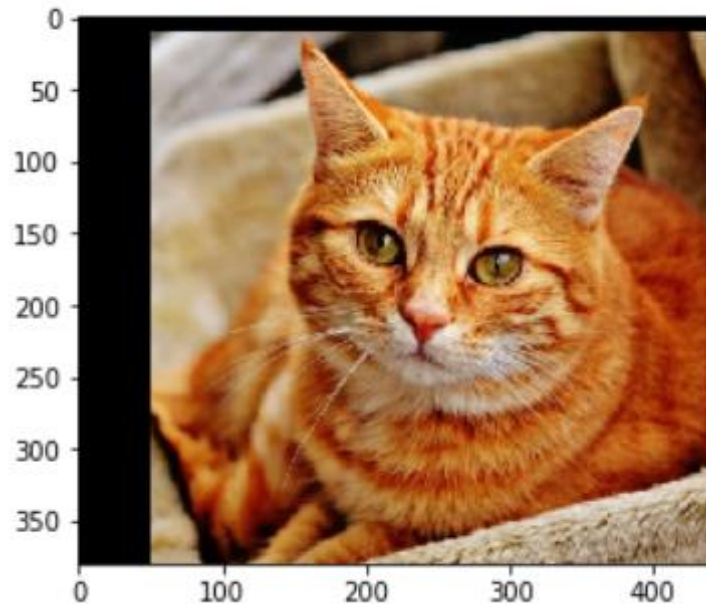
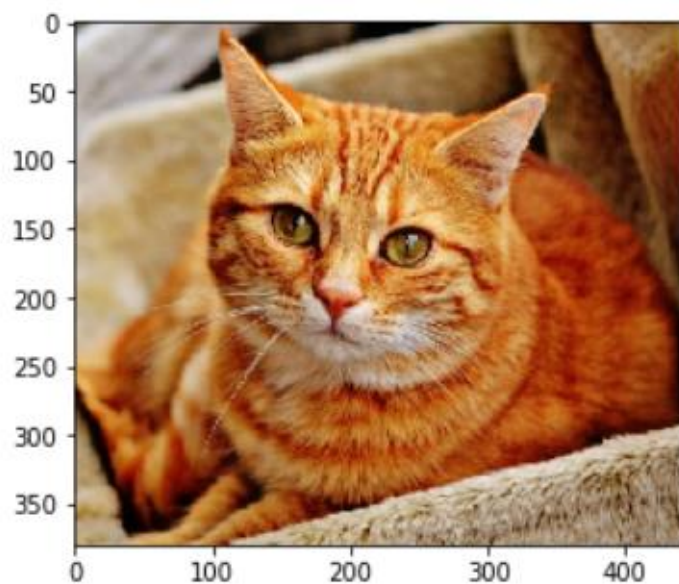
image = cv2.imread('image.png')
cv2.imshow('Image', image)
cv2.waitKey(0)

expand = cv2.resize(image, None, fx=2.0, fy=2.0, interpolation=cv2.INTER_CUBIC)
cv2.imshow('Image', expand)
cv2.waitKey(0)

shrink = cv2.resize(image, None, fx=0.8, fy=0.8, interpolation=cv2.INTER_AREA)
cv2.imshow('Image', shrink)
cv2.waitKey(0)
```

OpenCV 이미지 변형

이미지 위치 변경



OpenCV 이미지 변형

이미지 위치 변경

<code>cv2.warpAffine(image, M, dsize)</code>	이미지의 위치를 변경합니다.
--	-----------------

- M: 변환 행렬
- dsize: Manual Size

OpenCV 이미지 변형

변환 행렬과 변환

- 변환 행렬은 다음과 같은 형태로 정의됩니다.

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \end{bmatrix}$$

- 이미지의 모든 좌표 (a, b)는 다음의 좌표로 이동됩니다.

$$(M_{11} * a + M_{12} * b + M_{13}, M_{21} * a + M_{22} * b + M_{23})$$

OpenCV 이미지 변형

이미지 위치 변경

- 단순히 이미지의 위치를 변경할 때 변환 행렬은 다음과 같습니다.

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

- 이미지의 모든 좌표 (a, b)는 다음의 좌표로 이동됩니다.

$$(a + t_x, b + t_y)$$

OpenCV 이미지 변형

이미지 위치 변경

```
import cv2
import numpy as np

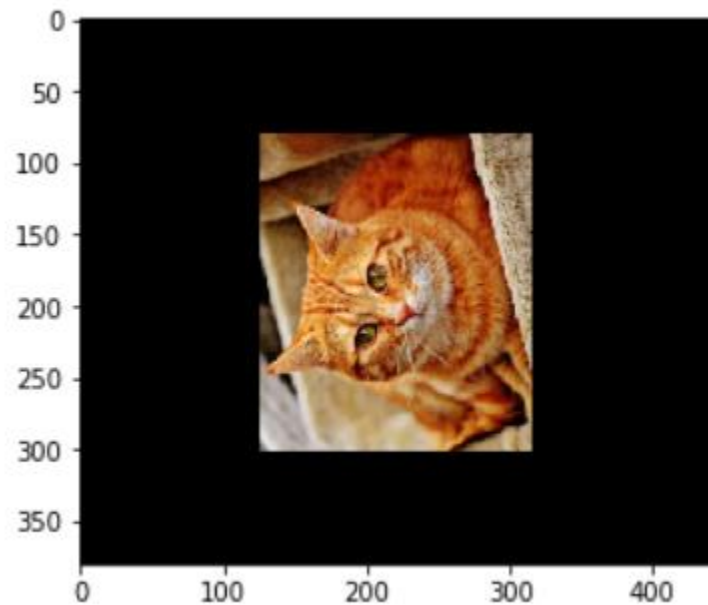
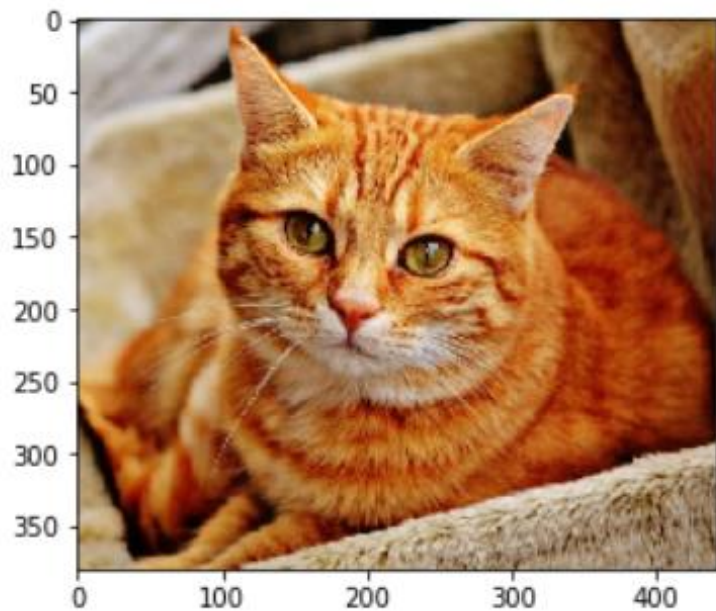
image = cv2.imread('image.png')

# 행과 열 정보만 저장합니다.
height, width = image.shape[:2]

M = np.float32([[1, 0, 50], [0, 1, 10]])
dst = cv2.warpAffine(image, M, (width, height))
cv2.imshow('Image', dst)
cv2.waitKey(0)
```

OpenCV 이미지 변형

이미지 회전



OpenCV 이미지 변형

이미지 회전

- 회전 변환을 위한 기본적인 행렬은 다음과 같습니다.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- 이 때 무게 중심을 적용할 수 있는 회전 변환 식은 다음과 같이 정의할 수 있습니다.

$$\alpha = scale * \cos\theta, \beta = scale * \sin\theta \text{ 일 때}$$

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) * center.x - \beta * center.y \\ -\beta & \alpha & \beta * center.x + (1 - \alpha) * center.y \end{bmatrix}$$

OpenCV 이미지 변형

이미지 회전

<code>cv2.getRotationMatrix2D(center, angle, scale)</code>	이미지 회전을 위한 변환 행렬을 생성합니다.
--	--------------------------

- center: 회전 중심
- angle: 회전 각도
- scale: Scale Factor

OpenCV 이미지 변형

이미지 회전

```
import cv2

image = cv2.imread('image.png')

# 행과 열 정보만 저장합니다.
height, width = image.shape[:2]

M = cv2.getRotationMatrix2D((width / 2, height / 2), 90, 0.5)
dst = cv2.warpAffine(image, M, (width, height))
cv2.imshow('Image', dst)
cv2.waitKey(0)
```