

Lab3_2013755_강단이

1.1 Interpolation

세부 기능 설명

확대시킬 가로와 세로의 비율을 ratioX로 5배, ratioY는 3배로 지정해줬다. 원본 해상도의 비율보다 가로로 5배, 세로로 3배 크게 값을 저장할 dst를 선언해준다.

크게 확대될 픽셀들에서 양선형 보간법을 적용해준다. 확대된 이미지의 좌표 xd, yd에 대응하는 원본 이미지의 좌표를 소수로 변환한 값은 (xo, yo)이다.

(xo, yo)좌표 주변의 네 픽셀 좌표를 찾기 위한 계산을 하기 위해 math.floor와 math.ceil을 4개의 픽셀 좌표를 위한 x, y값을 찾아준다. Xleft는 xo의 내림, xright는 xo의 올림, yup은 yo의 내림, ydown은 yo의 올림을 해준 값이다. 보간을 위한 가중치 계산을 위해 알파와 베타 값을 구했고, 이는 xo와 xleft의 차, yo와 yup의 차이로 구했다. 만일 올림한 값이 255를 넘는다면 255로 지정하게 조건문을 걸어줬다.

확대된 이미지의 픽셀값을 계산하기 위해 보간 공식을 적용한다.

$(1-a) * (1-b) * \text{src}[yup, xleft]$: 좌측 상단 픽셀의 기여도,

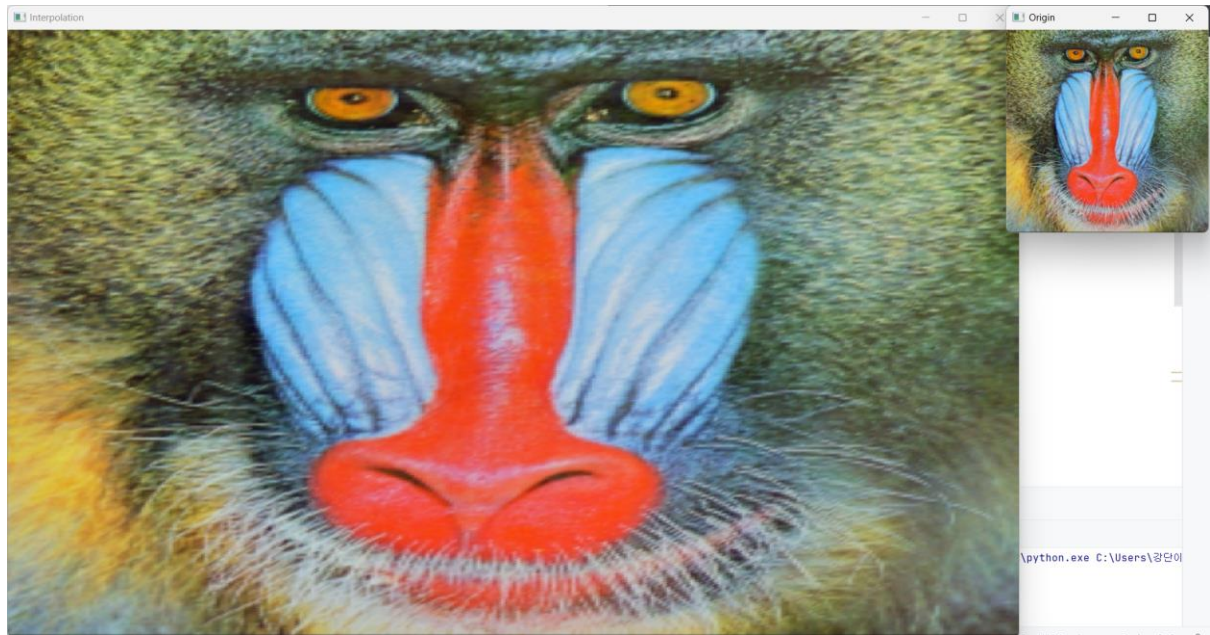
$a * (1-b) * \text{src}[yup, xright]$: 우측 상단 픽셀의 기여도,

$(1-a) * b * \text{src}[ydown, xleft]$: 좌측 하단 픽셀의 기여도,

$a * b * \text{src}[ydown, xright]$: 우측 하단 픽셀의 기여도를 의미한다.

적용해 나온 값을 dst에 저장해 출력한다.

기능별 실행 화면 캡처



가로 5배, 세로 3배 된 이미지와 우측 상단의 본래의 이미지

소스코드

```

import cv2 as cv
import numpy as np
import math

src = cv.imread("Mandrill.bmp", cv.IMREAD_COLOR)
H, W, C = src.shape[:]

ratioX = 5
ratioY = 3

dst = np.zeros((H*ratioY, W*ratioX, C), src.dtype)

cols = H
rows = W

for yd in range (ratioY * cols):
    for xd in range (ratioX * rows):
        for c in range (3):
            xo = xd / ratioX
            yo = yd / ratioY

            xleft = math.floor(xo)
            xright = math.ceil(xo)
            yup = math.floor(yo)
            ydown = math.ceil(yo)

            a = xo - xleft
            b = yo - yup

            if (xright > 255): xright = 255
            if (ydown > 255): ydown = 255
            if (xleft < 0): xleft = 0
            if (yup < 0): yup = 0
            dst[yd, xd] = (1-a) * (1-b) * src[yup, xleft] + a * (1-b) * src[yup, xright]
+ (1-a) * b * src[ydown, xleft] + a * b * src[ydown, xleft]

cv.imshow("Interpolation", dst)
cv.imwrite("./Interpolation.bmp", dst)
cv.waitKey(0)

```