

Open_cv02


영상하수문 이용하 영상퍼지과 차원 변화
L L M 수문 이용하 영상퍼지과 차원 변화
L L M 수문 이용하 영상퍼지과 차원 변화

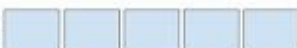
기본 배열(Array) 처리 함수

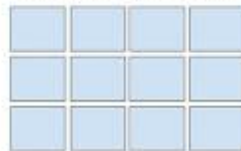
- 파이썬에서는 배열을 처리하기 위한 자료형
 - 열거형(sequence) 객체 - 리스트, 튜플, 사전(dictionary)
- 명칭 표현
 - 1차원 데이터 - 벡터
 - 2차원 데이터 - 행렬
 - 1차원과 2차원 데이터 통칭해서 배열
 - 추가적인 3차원

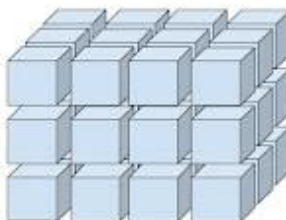
기본 배열(Array) 처리 함수

Rank	Math entity
0	Scalar (magnitude only)
1	Vector (magnitude and direction)
2	Matrix (table of numbers)
3	3-Tensor (cube of numbers)
n	n-Tensor (you get the idea)

Rank 0: 
(scalar)

Rank 1: 
(vector)

Rank 2: (matrix)


Rank 3: 

(11)

5 3 7

SCALAR

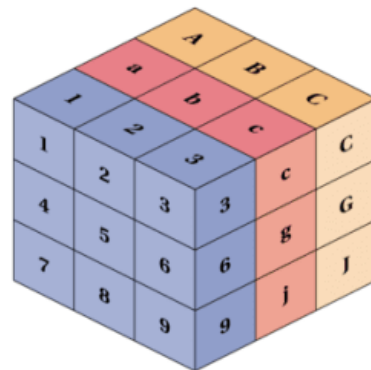
Row Vector
(shape 1x3)

5
1.5
2

Column Vector
(shape 3x1)

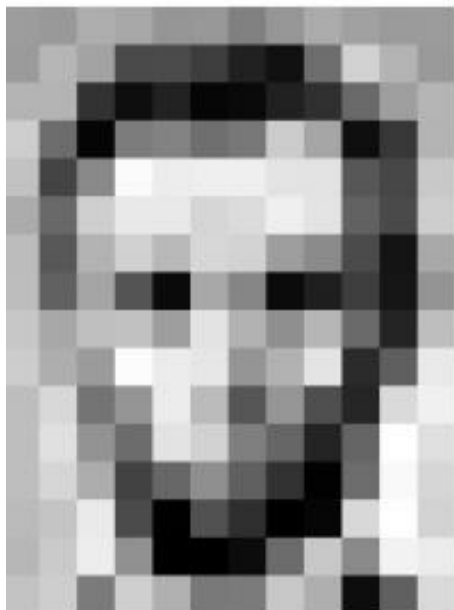
$\begin{bmatrix} 4 & 19 & 8 \\ 16 & 3 & 5 \end{bmatrix}$

MATRIX



TENSOR

기본 배열(Array) 처리 함수



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	236	76	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	236	76	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

<https://forum.processing.org/>

기본 배열(Array) 처리 함수

• 기본 배열 처리 함수

함수 설명

`cv2.flip(src, flipCode[, dst]) → dst`

■ 설명: 입력된 2차원 배열을 수직, 수평, 양축으로 뒤집는다.

인수 설명	■ src, dst	입력 배열, 출력 배열
	■ flipCode	배열을 뒤집는 축
	- 0	x축을 기준으로 위아래로 뒤집는다.
	- 1	y축을 기준으로 좌우로 뒤집는다.
	- -1	양축(x축, y축 모두)을 기준으로 뒤집는다.

`cv2.repeat(src, ny, nx[, dst]) → dst`

■ 설명: 입력 배열의 반복된 복사본으로 출력 배열을 채운다.

인수 설명	■ src, dst	입력 배열, 출력 배열
	■ ny, nx	수직 방향, 수평방향 반복 횟수

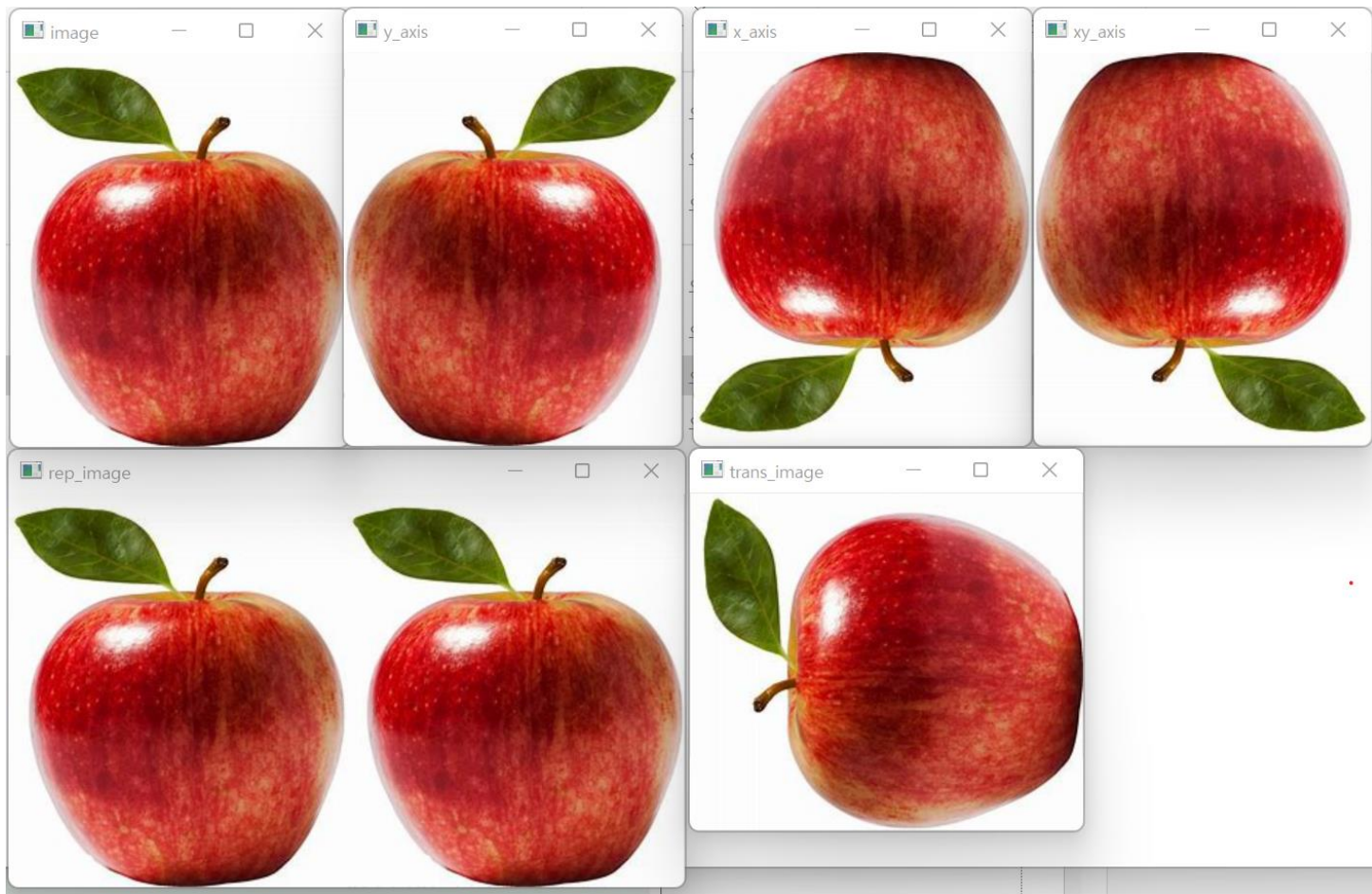
`cv2.transpose(src[, dst]) → dst`

■ 설명: 입력 행렬의 전치 행렬을 출력으로 반환한다.

인수 설명	■ src, dst	입력 배열, 출력 배열
----------	------------	--------------

$$M = \begin{bmatrix} 2 & -9 & 3 \\ 13 & 11 & -17 \\ 3 & 6 & 15 \\ 4 & 13 & 1 \end{bmatrix} \quad M^T = \begin{bmatrix} 2 & 13 & 3 & 4 \\ -9 & 11 & 6 & 13 \\ 3 & -17 & 15 & 1 \end{bmatrix}$$

행렬처리함수 실행 결과



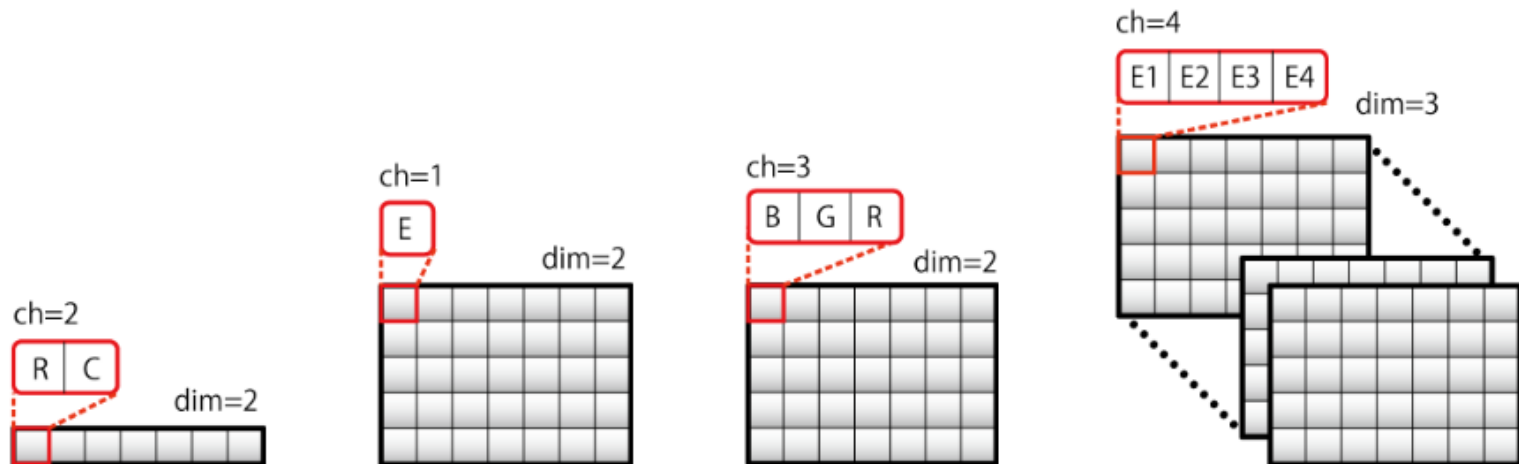
채널 처리 함수

- 채널 개념

- 채널: 요소의 차원

- 차원: 여러 요소로 구성된 배열의 차원

다채널 다차원 배열의 예(Dim=2 Ch=2, dim=2 Ch=1, Dim=2 Ch=3, dim=3 Ch=4):



RGB는 비디오 기술에 사용되는 컴포넌트 비디오 신호에 사용

잠깐 ! 가산, 감산 , 스칼라 곱

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 1+4 \\ 2+5 \\ 3+6 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 9 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} + \begin{bmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} = \begin{bmatrix} 1+7 & 2+8 & 3+9 \\ 4+10 & 5+11 & 6+12 \end{bmatrix} =$$

$$10 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 10 * 1 \\ 10 * 2 \\ 10 * 3 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix}$$

$$10 \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 10 * 1 & 10 * 2 & 10 * 3 \\ 10 * 4 & 10 * 5 & 10 * 6 \end{bmatrix} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

내적 (inner product)

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = 1 \times 4 + 2 \times 5 + 3 \times 6 = 32$$

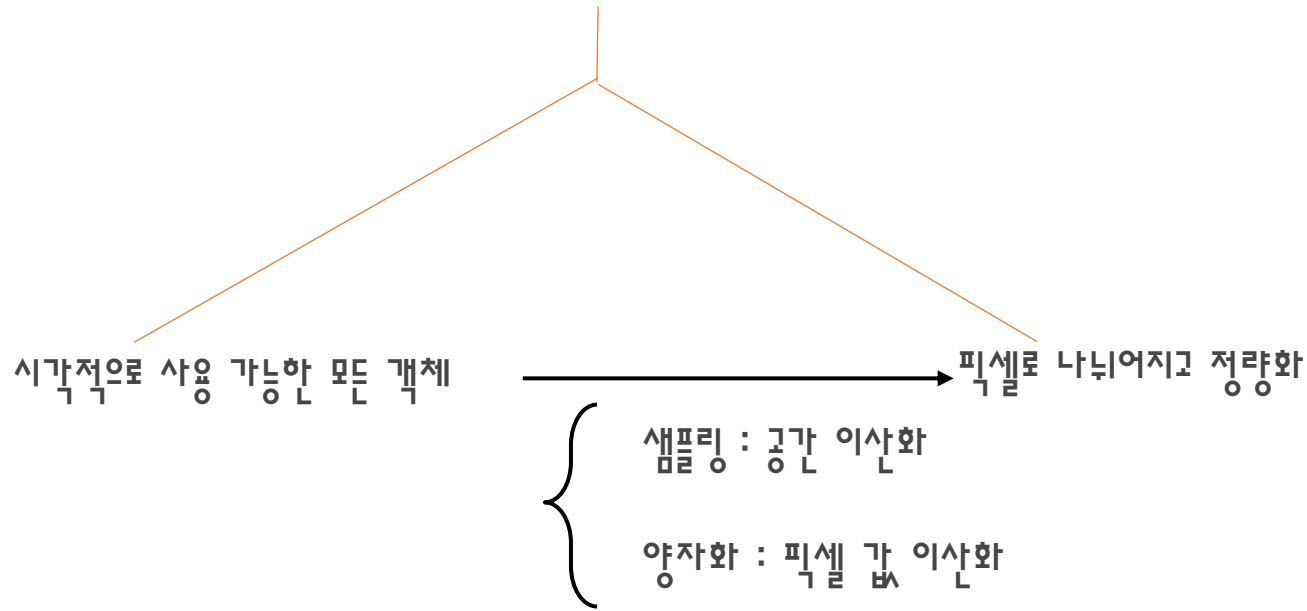
채널 처리 함수

- 회색조는 회색의 그라디언트로, 흰색과 검정색 사이의 중간 색상이며 254 음영으로 표현.
- 흑백과는 달리 풍부한 그라데이션으로 표현되므로 사용된다.
- 일반적인 테스트 중시 문서인 경우 흑백으로 ~.
- 바탕에 사진과 일러스트레이션을 사용할 때 가능한 한 생생하게 표현하려면 회색조를 선택

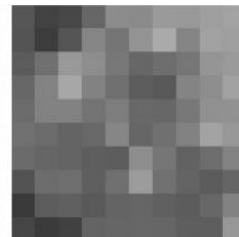
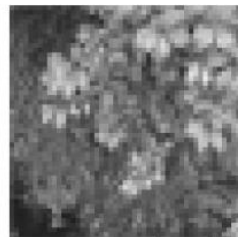
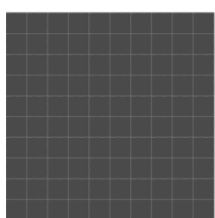


채널 처리 함수

. 디지털 이미지와 아날로그 이미지란?



채널 처리 함수



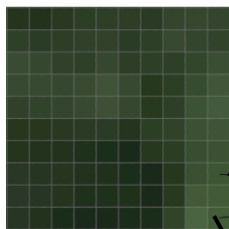
300* 300

100*100

10* 50

10*10

두개의 채널 : 정량화 -> 많은 수의 사각형 (공간 연속데이터 픽셀)



색 깊이, 색 깊이, 밀도 값, 농도 수준, 회색 수준 = 색상 그라데이션을 얼마나 나눌까?

(62,85,51)
R G B

(68,92,56)
R G B



$$2^8 = 2*2*2*2*2*2*2*2 = 256$$

채널 처리 함수

• 디지털 이미지 소싱은 ?

- 픽셀 데이터 + 이미지 데이터 이외의 데이터 + 헤더 및 메타 데이터가 포함
- 파일 이름
- 생성 날짜, 업데이트 날짜 및 시간, 액세스 날짜 및 시간
- 이미지 해상도
- 색상
- 압축 유형
- 파일의 구조 등을 나타내는 데 필요한 기타 데이터

채널 처리 함수

• 채널 관련 함수

함수 설명

`cv2.merge(mv[, dst]) → dst`

■ 설명: 여러 개의 단일채널 배열을 다채널 배열로 합성한다.

인수	■ mv	합성될 입력 배열 혹은 벡터, 합성될 단일채널 배열들의 크기와 깊이(depth)가 동일해야 함
설명	■ dst	입력 배열과 같은 크기와 같은 깊이의 출력 배열

`cv2.split(m[, mv]) → mv`

■ 설명: 다채널 배열을 여러 개의 단일채널 배열로 분리한다.

인수	■ m	입력되는 다채널 배열
설명	■ mv	분리되어 반환되는 단일채널 배열들의 벡터

통계 관련 함수

- `reduce()` 함수 감축 방향

dim=0 : 한 행으로 감축

11	2	3	4	10
6	10	15	9	7
7	12	8	14	1



24	24	26	27	18
----	----	----	----	----

rtype =
cv2.REDUCE_SUM

dim=1 : 한 열로 감축

11	2	3	4	10
6	10	15	9	7
7	12	8	14	1



6
9.4
8.4

rtype =
cv2.REDUCE_AVG

통계 관련 함수

cv2.SORT_EVERY_ROW

53	90	23	73	61
35	33	79	68	30
54	99	21	62	73

행 단위 정렬
+ 오름차순

m_sort1

23	53	61	73	90
30	33	35	68	79
21	54	62	73	99

sort() 주요 속성

cv2.SORT_EVERY_COLUMN

53	90	23	73	61
35	33	79	68	30
54	99	21	62	73

열단위 정렬+오름차순

m_sort3

35	33	21	62	30
53	90	23	68	61
54	99	79	73	73

cv2.SORT_EVERY_ROW + cv2.SORT_DESCENDING

53	90	23	73	61
35	33	79	68	30
54	99	21	62	73

행 단위 정렬
+ 내림차순

m_sort2

90	73	61	53	23
79	68	35	33	30
99	73	62	54	21

행렬 연산 함수

함수 설명

`cv2.gemm(src1, src2, alpha, src3, beta[, dst[, flags]]) → dst`

■ 설명: 일반화된 행렬 곱셈을 수행한다.

■ 수식: $dst = \alpha \cdot src1^T \cdot src2 + \beta \cdot src3^T$

인수
설명

- src1, src2 행렬 곱을 위한 두 입력 행렬(np.float32/np.float64형 2채널까지 가능)
- alpha 행렬 곱($src1^T \cdot src2$)에 대한 가중치
- src3 행렬 곱($src1^T \cdot src2$)에 더해지는 델타 행렬
- beta src3 행렬에 곱해지는 가중치
- dst 출력 행렬
- flags 연산 플래그 - 옵션을 조합하여 입력 행렬들을 전치

옵션	값	설명
cv2.GEMM_1_T	1	src1을 전치
cv2.GEMM_2_T	2	src2을 전치
cv2.GEMM_3_T	4	src3을 전치

`cv2.perspectiveTransform(src, m[, dst]) → dst`

■ 설명: 입력 벡터들에 대해서 투영(perspective) 변환 m을 수행한다.

■ 수식: 3차원 좌표인 경우

$$(x, y, z) \rightarrow (x'/w, y'/w, z'/w)$$

여기서

$$(x', y', z', w') = m \cdot [x, y, z, 1] \quad w = \begin{cases} w' & \text{if } w' \neq 0 \\ \infty & \text{otherwise} \end{cases}$$

인수
설명

- src 좌표로 변환될 2채널 혹은 3채널의 부동소수점 배열
- dst src와 같은 크기와 타입의 출력 배열
- m 3×3 혹은 4×4 부동소수점의 투영 변환 행렬

행렬 연산 함수

이미지 밝은 어둡게 조정

- 이미지/비디오 입출력 장비의 특성을 나타내는 인덱스-> **감마 값(γ 값)**
- 감마 값은 입출력 장치의 신호와 픽셀의 밝기 간의 관계를 나타낸다.
- 입출력 장치가 이미지 또는 비디오를 출력할 때 감마 값을 보정하여 밝고 어두운 색상이 올바르게 출력될 수 있도록 감마 보정을 한다.

$$src1^T \cdot src2 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}^T \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 5 & 7 & 9 \\ 10 & 14 & 18 \\ 15 & 21 & 27 \end{bmatrix}$$

$$src1 \cdot src2^T = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 14 & 32 \\ 14 & 32 \end{bmatrix}$$

$$src1 \cdot src3 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 12 \\ 6 & 12 \end{bmatrix}$$

감마 처리된 연산 결과