MMU data format 분석

염지현, 김형범

지현 & 형범 의견

: MMU 담당자분께 넘겨받은 연산 및 데이터 format은 현재 구현한 sparse matrix와 거리가 있기 때문에 이 방식이 우리와 맞지 않다고 생각..

Data Format 분석

Data format 분석

파일 이름	용도	크기	비고
Image_pixel	256*256 image pixel 값 한 줄로 펴서 저장	256*256	Lenna image 기준
Img_pruning	임계치 이하 값을 0으로 치환하는 압축 방법	37594	
cifm	이미지를 3*3*128로 저장	3*3*129 → 1*1*1161	
Cimf_value	CIFM에서 0이 아닌 value 값을 저장	774	
cifm_pos	CIFM에서 0이 아닌 값의 위치 저장	774	
Ckm	커널을 3*3*128로 저장	3*3*129 → 1*1*1161	
Ckm_value	ckm에서 0이 아닌 value 값을 저장	768	
Ckm_pos	ckm에서 0이 아닌 값의 위치 저장	768	
Same_pos	Cifm과 ckm position에서 같은 position 값 저장	511	
Mac_out	Same_pos 연산 결과값 저장	511	

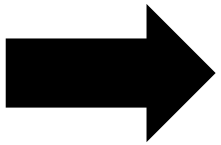
Data format 분석

파일 이름	용도	크기(저장된 숫자 개수)	비고
Image_pixel	256*256 image pixel 값 한 줄로 펴서 저장	256*256	Lenna image 기준
Img_pruning	임계치 이하 값을 0으로 치환하는 압축 방법	37594	
cifm	이미지를 3*3*128로 저장	3*3*129 → 1*1*1161	
Cimf_value	CIFM에서 0이 아닌 value 값을 저장	774	
cifm_pos	CIFM에서 0이 아닌 값의 위치 저장	774	
Ckm	커널을 3*3*128로 저장	3*3*129 → 1*1*1161	
Ckm_value	ckm에서 0이 아닌 value 값을 저장	768	
Ckm_pos	ckm에서 0이 아닌 값의 위치 저장	768	
Same_pos	Cifm과 ckm position에서 같은 position 값 저장	511	
Mac_out	Same_pos 연산 결과값 저장	511	

Data format 분석

1) Image_pixel: RGB → YCbCr 로 변환하여 저장

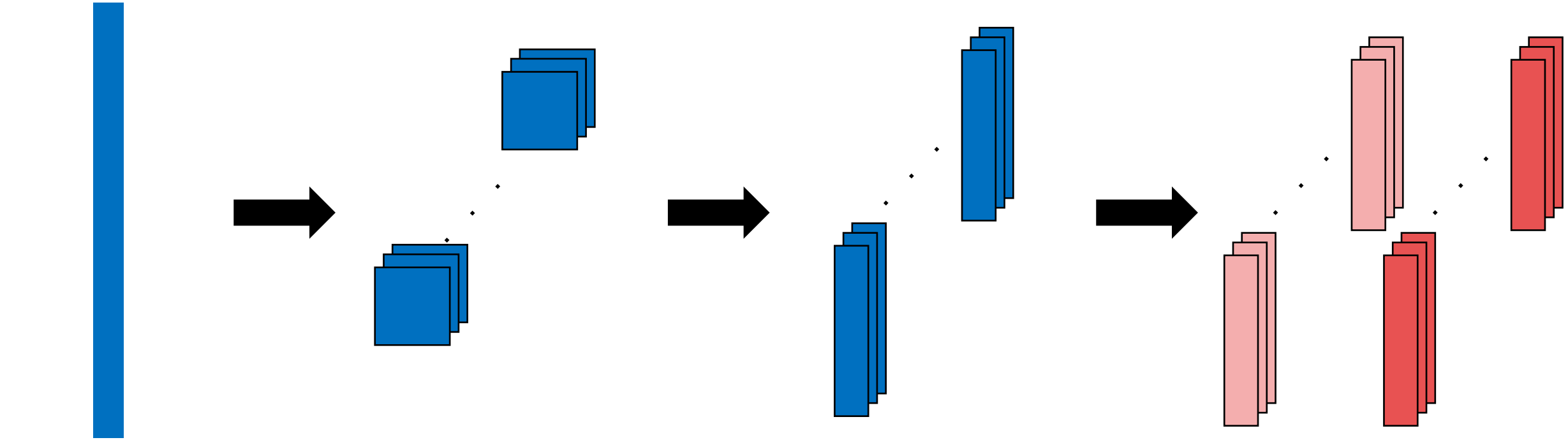
1920*1080



1*1*2,073,600

Data format 분석

3) CIFM: 이미지를 3*3*128로 저장



1*1*2,073,600*128
(channelization)

3*3*128*230,400

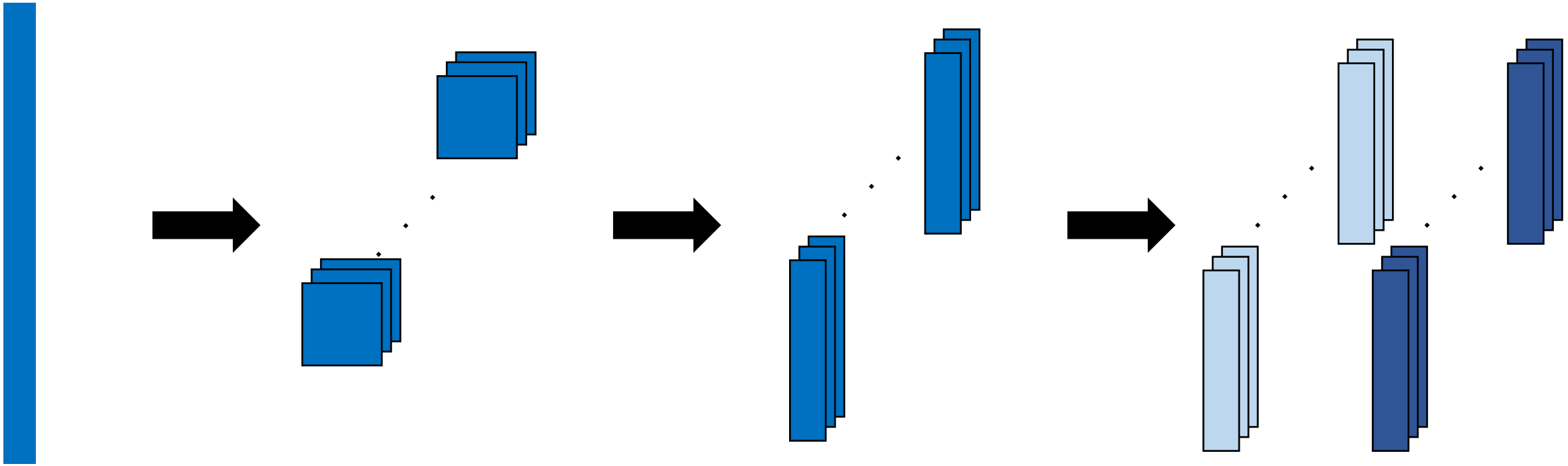
1*1*1152*230,400

1*1*n*230,400*2(val, pos)
n: # of non-zero elements

Data format 분석

4) CKM: 커널을 3*3*128로 저장

Input 개수: 12, output 개수: 12라고 가정



3*3*128*144
(channelization)

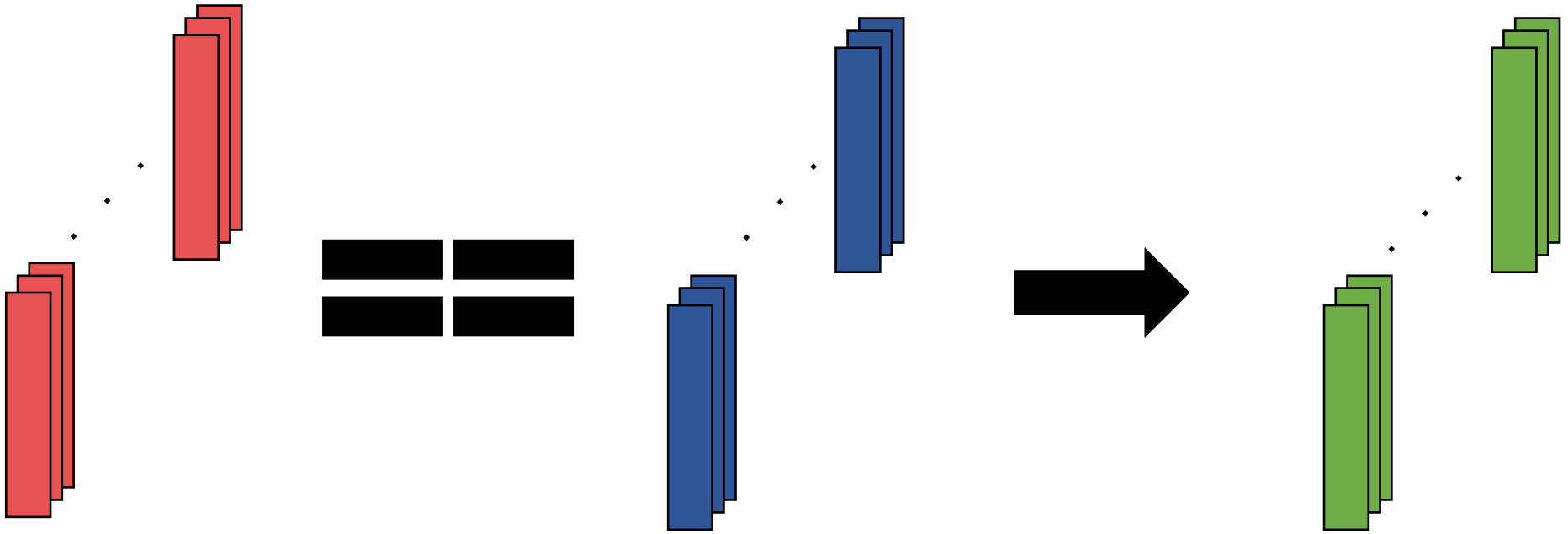
3*3*128*144

1*1*1152*144

1*1*1152*144*2(val, pos)
n: # of non-zero elements

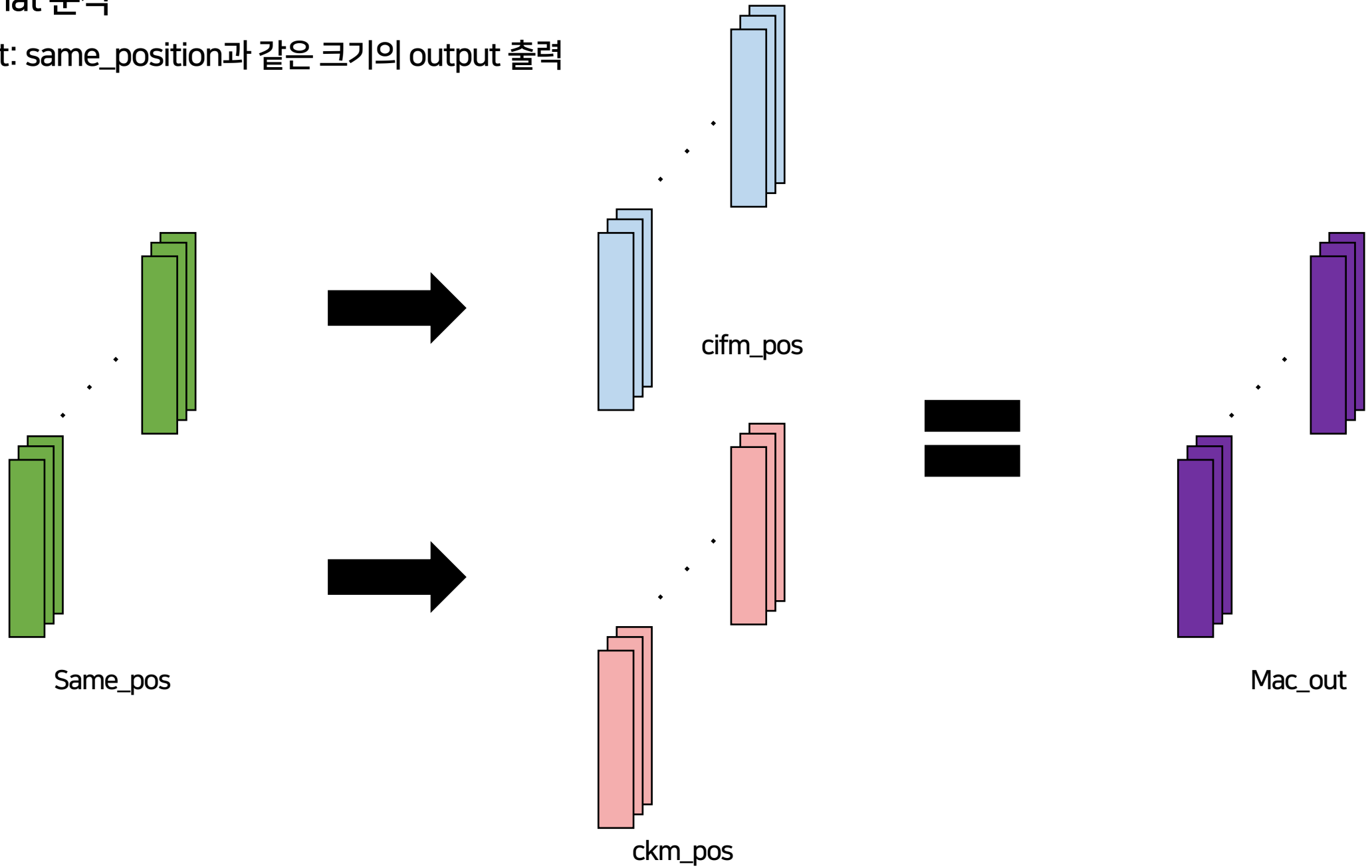
Data format 분석

5) same_position: CIFM, CKM position 같은 부분 저장



Data format 분석

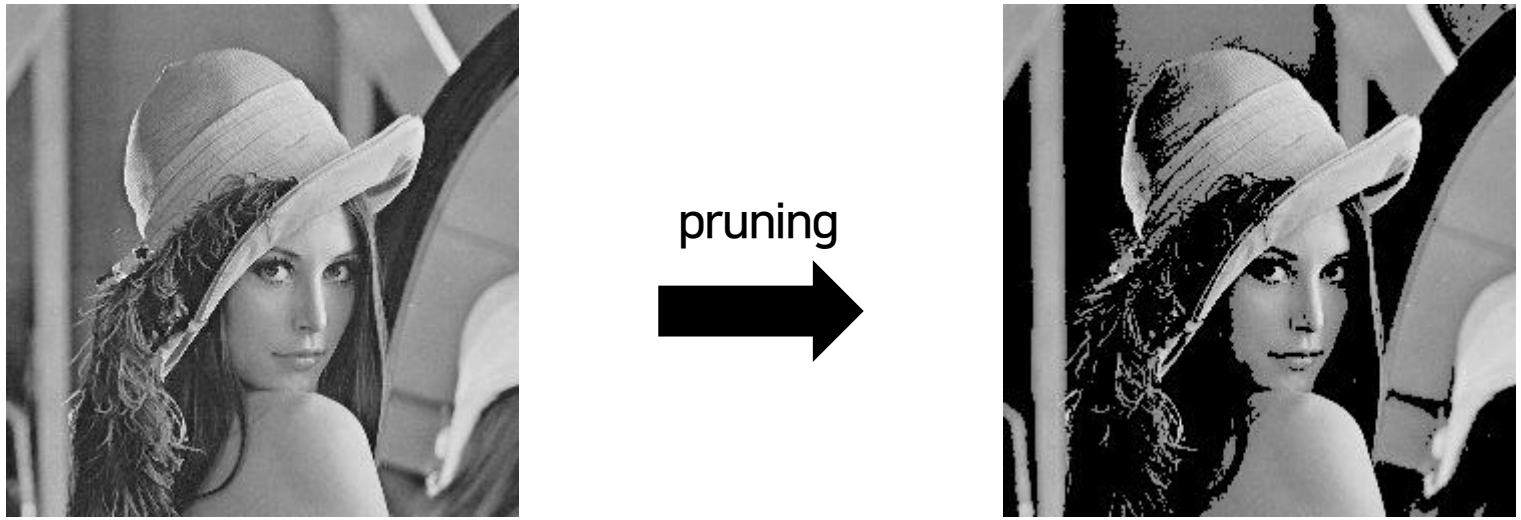
6) mac_out: same_position과 같은 크기의 output 출력



MMU 관점 FSRCNN 연산 문제점

MMU 관점 FSRCNN 연산 문제점

1) Pruning을 통해 이미지 압축 시 많은 정보 손실
: 머리카락처럼 살려야 하는 정보를 날린 상태에서 SR 진행 시 결과 보장 불가



MMU 관점 FSRCNN 연산 문제점

1) Pruning을 통해 이미지 압축 시 많은 정보 손실
: 머리카락처럼 살려야 하는 정보를 날린 상태에서 SR 진행 시 결과 보장 불가

SR 버전 sparse matrix

Original image

1
2
255
256

Image_pixel

									0	0	0	0
0										0	0	0
0	0										0	0
0	0	0										0
0	0	0	0									
									0	0	0	0
0										0	0	0
중략												
0	0	0	0									

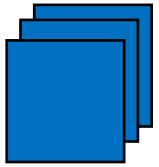
Sparse Matrix

MMU 버전 sparse matrix

Pruning image

1
2
n-1
n

Sparse Matrix

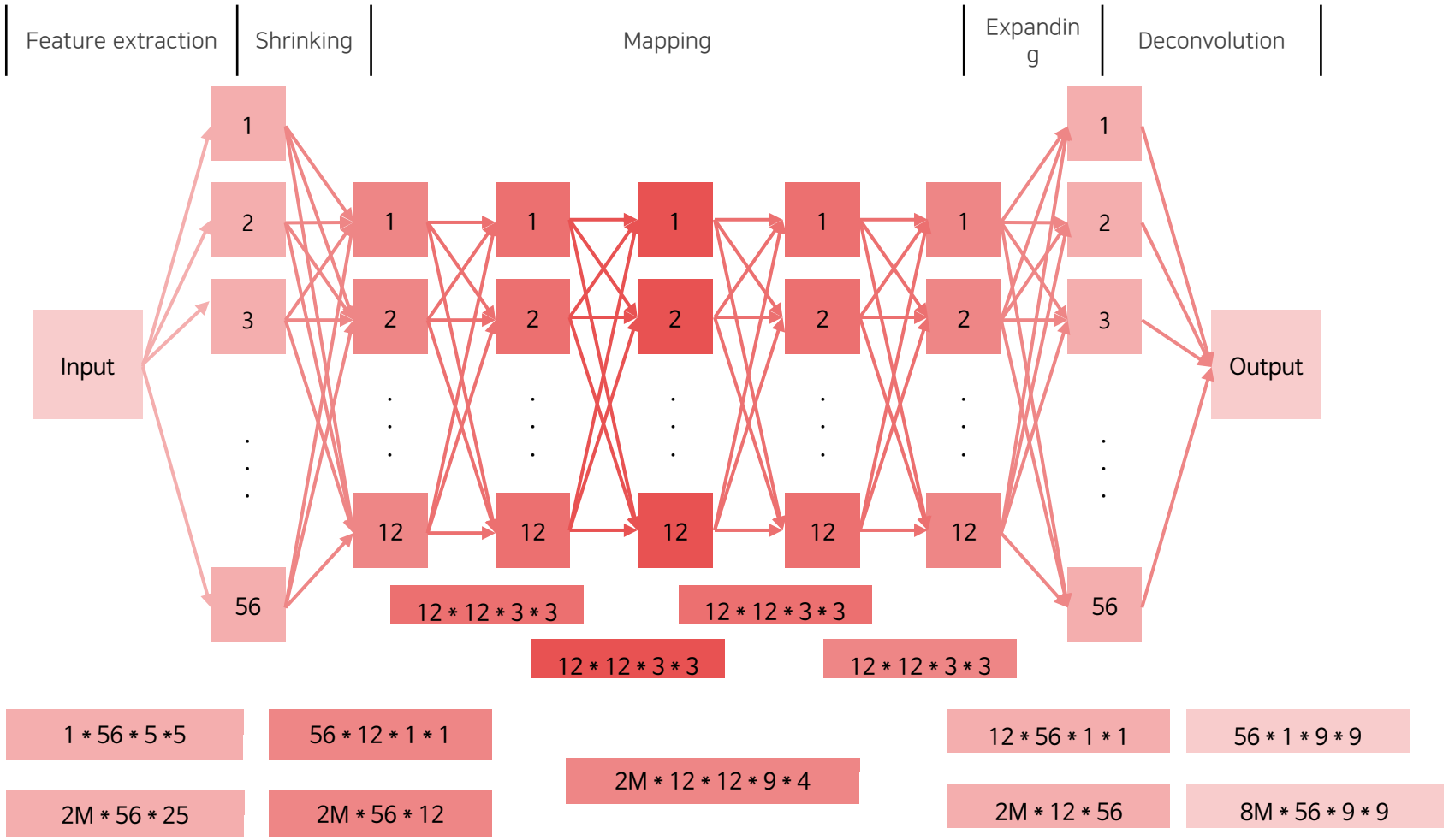


kernel

MMU 관점 FSRCNN 연산 문제점

2) 1*1, 3*3, 5*5, 9*9 kernel 사용

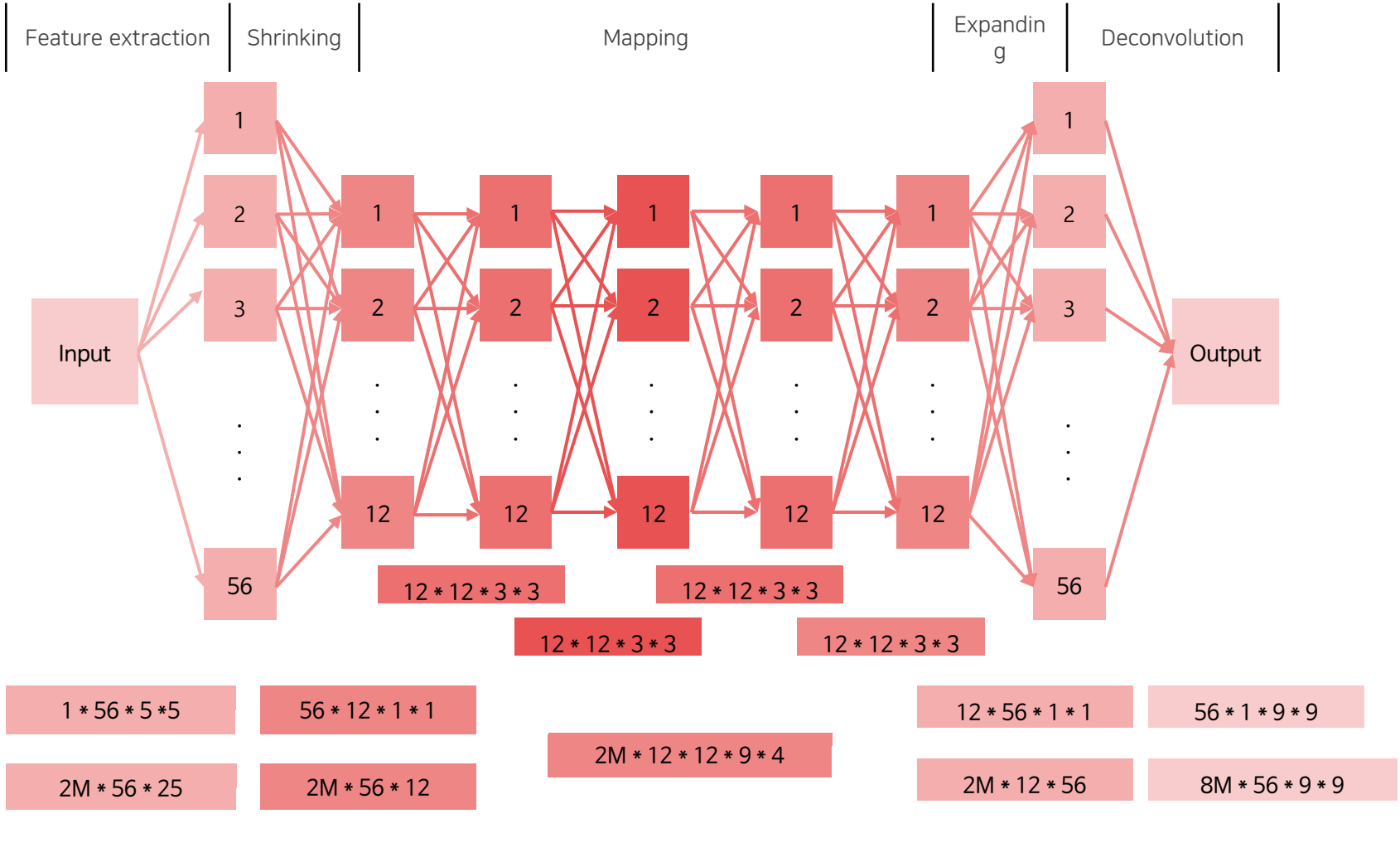
: 3*3 size 외에 다른 size 사용 가능성 파악 필요



MMU 관점 FSRCNN 연산 문제점

3) 연산량 증가

: layer 별로 (input 쪼개기 → channelization → 0이 아닌 val, pos 저장 → 연산 → 하나의 output으로 합침) 과정 반복



2M을 3*3 크기로 쪼갰을 때 231,736개 파일 생성

단계	Output featuremaps	연산 횟수
Feature extraction	2M * 56	56 * 5 * 5 = 1400
Shrinking	2M * 12	672 * 1 * 1 = 672
Mapping	2M * 12 * 4	(144 * 3 * 3) * 4 = 5184
Expanding	2M * 56	672 * 1 * 1 = 672
Deconvolution	8M * 1	56 * 9 * 9 = 4536