

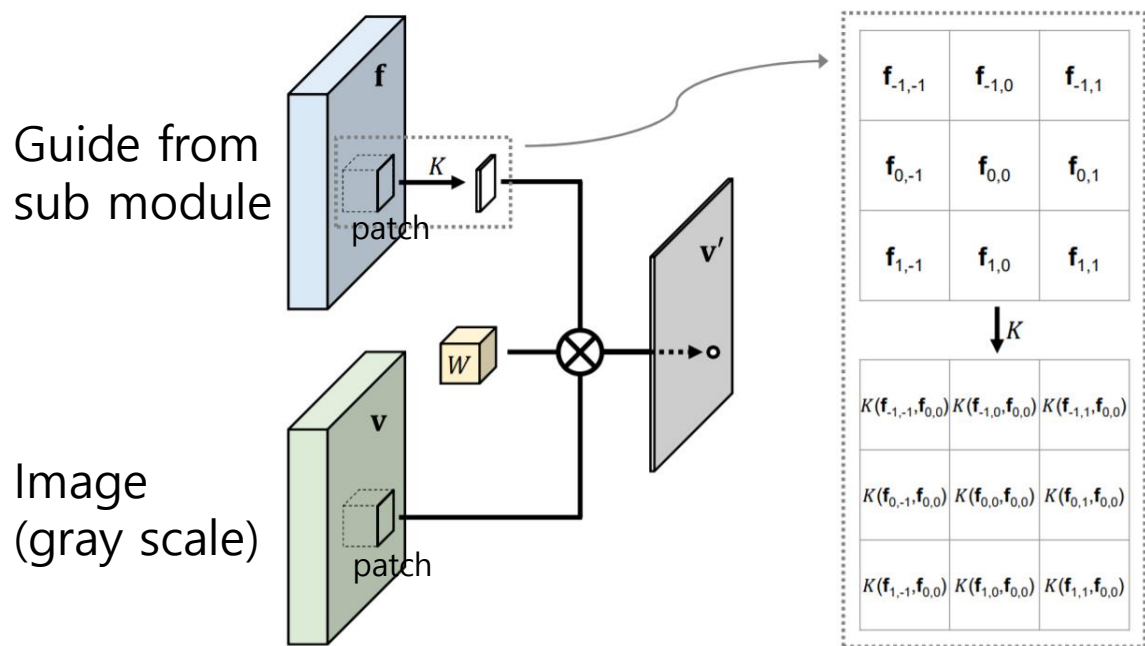
# PAC 모듈을 이용한 딥러닝 기반 스테레오 매칭

9

(리포트 용)

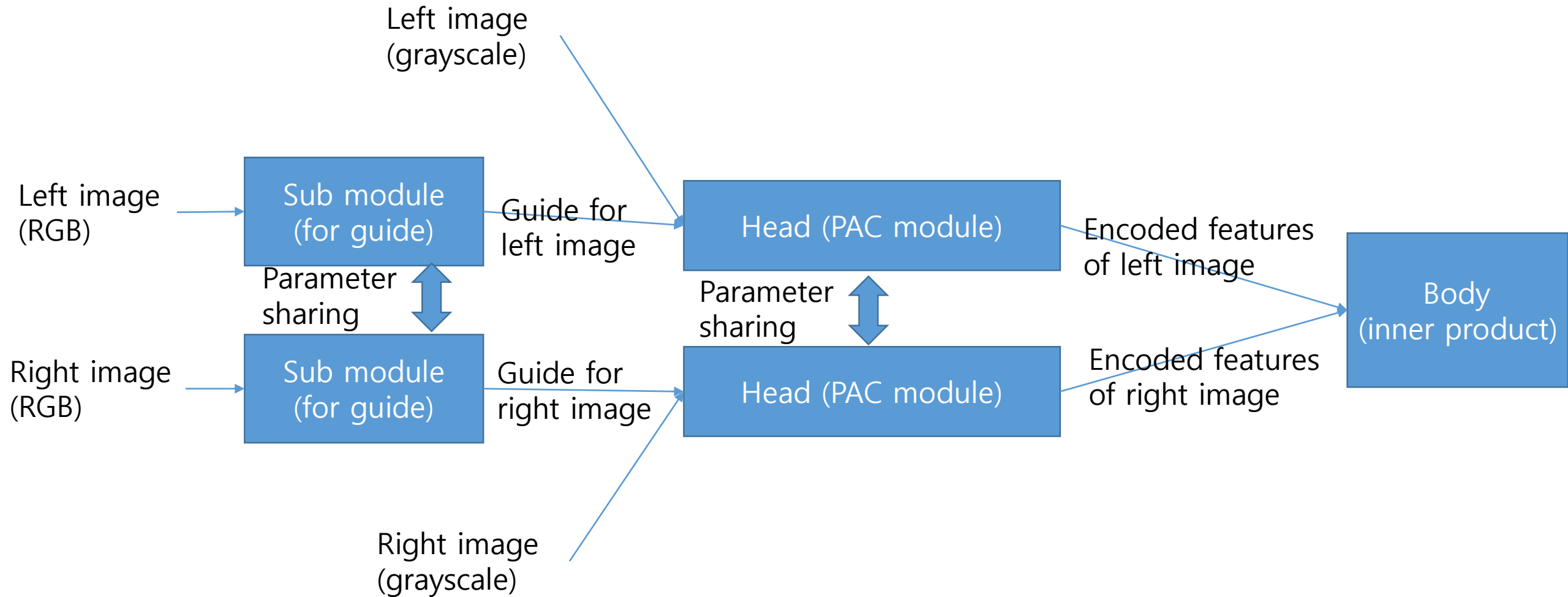
# 모듈과 모델 소개

# PAC 모듈

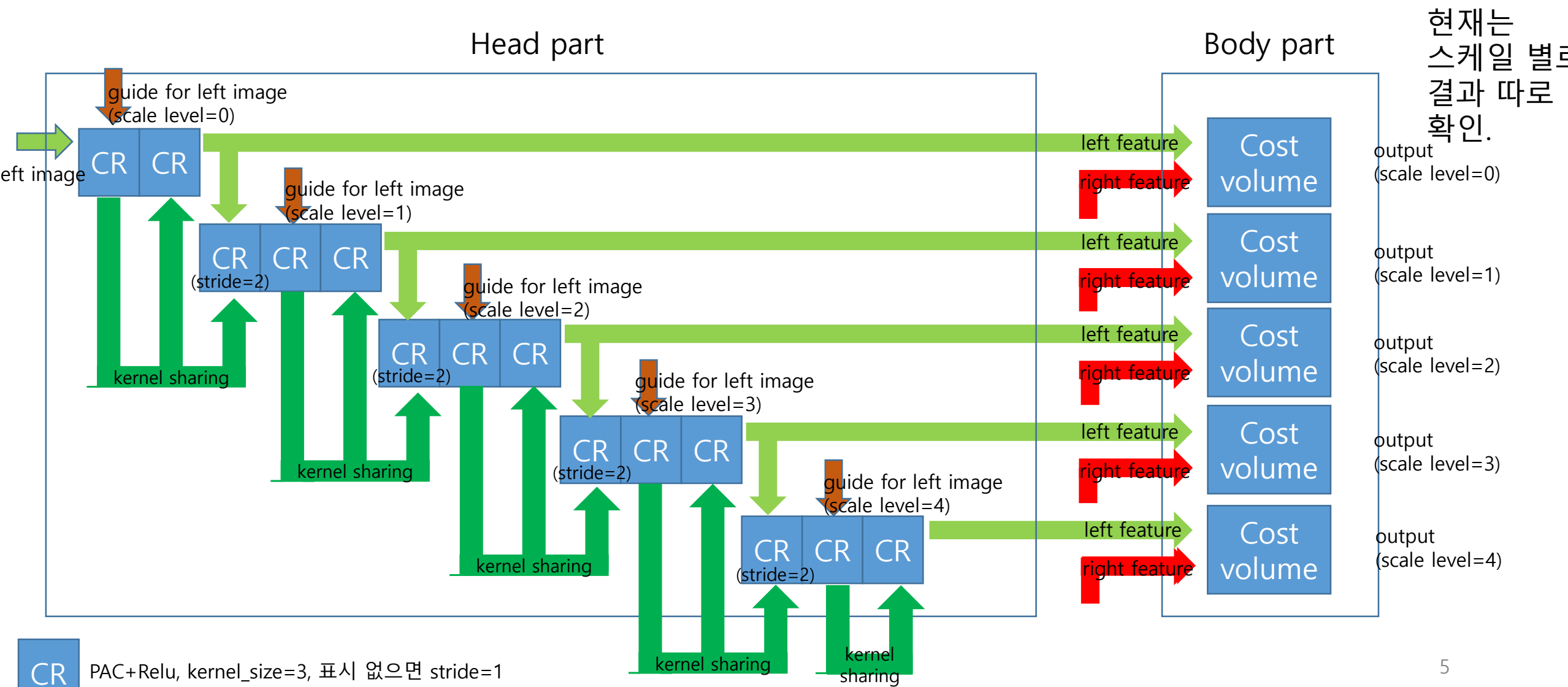


- Guide에서 뽑아낸 패치에서 정중앙에 위치한 픽셀의 feature 값과 다른 픽셀의 feature 값의 차이(l2 norm)를 가지고 kernel(=attention?) 생성
- 의도: 패치의 중앙에 위치한 픽셀과 다른 특성을 가진 픽셀은 무시하고 컨볼루션 연산을 진행함.

# 모델 구조



# 모델 구조 (Body 부분 미완성)



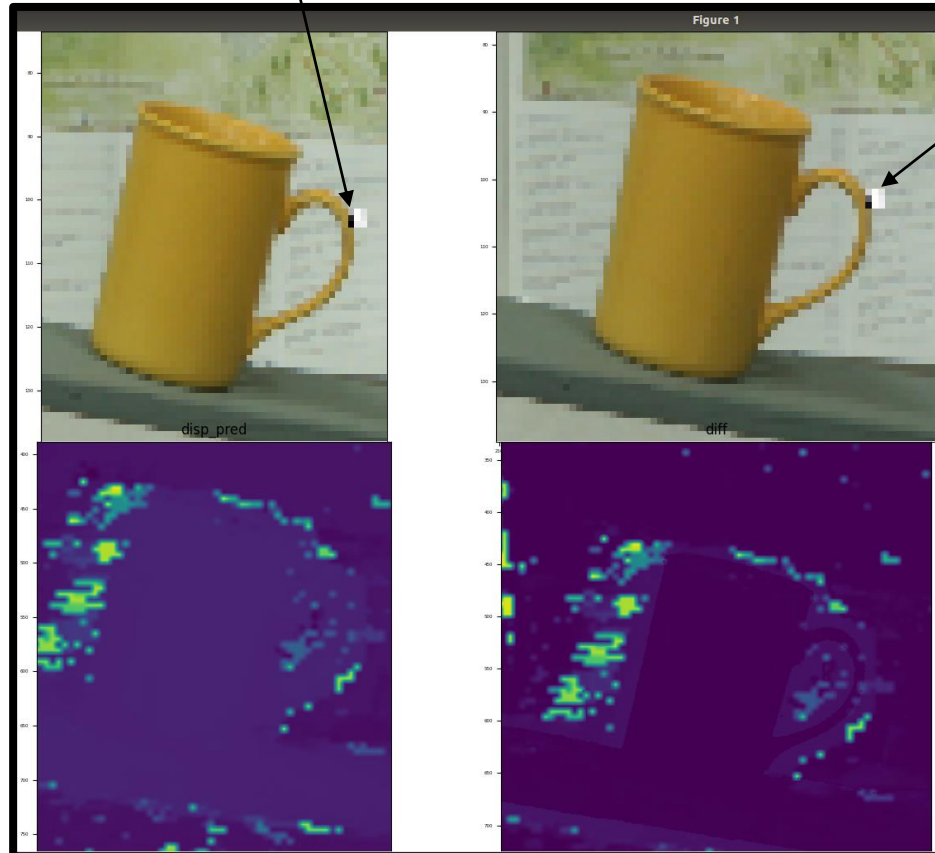
Stereo matching task에서  
PAC 모듈의 커널 적용 방식이  
갖는 문제점과 해결 방안

# 이미지 표현 설명

Left 이미지에서 선택된 픽셀과  
그 위의 커널

left 이미지에서 선택된 픽셀과 매칭될  
것으로 예상된 right 이미지의 픽셀과  
그 위의 커널

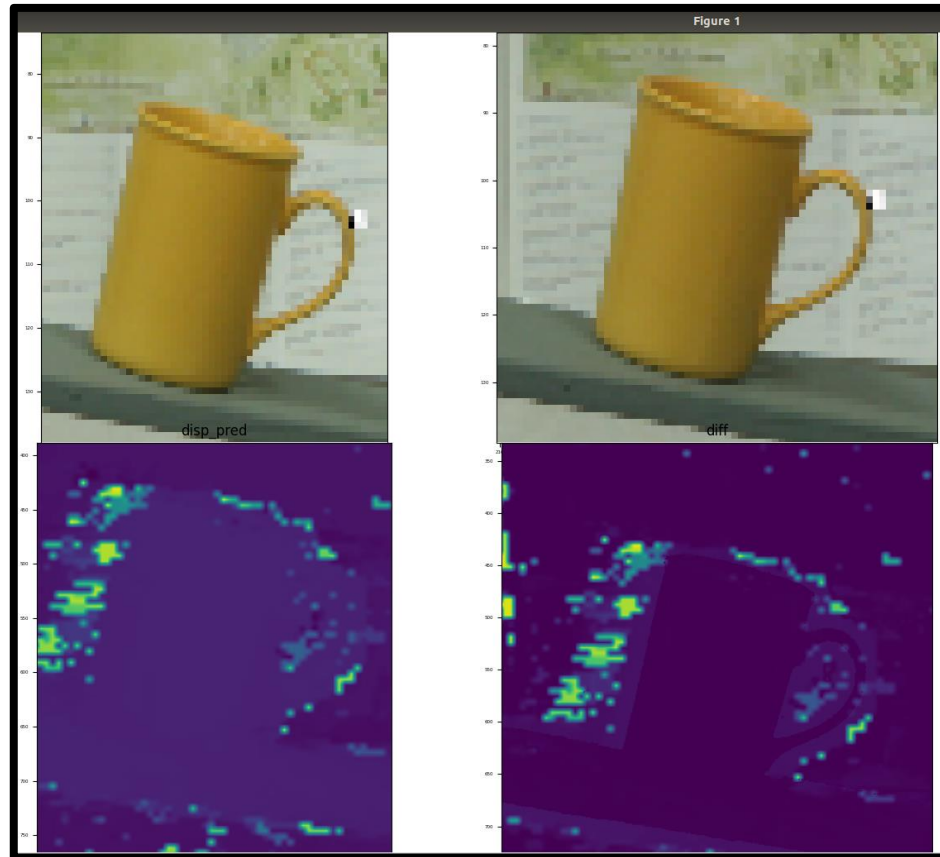
Left image



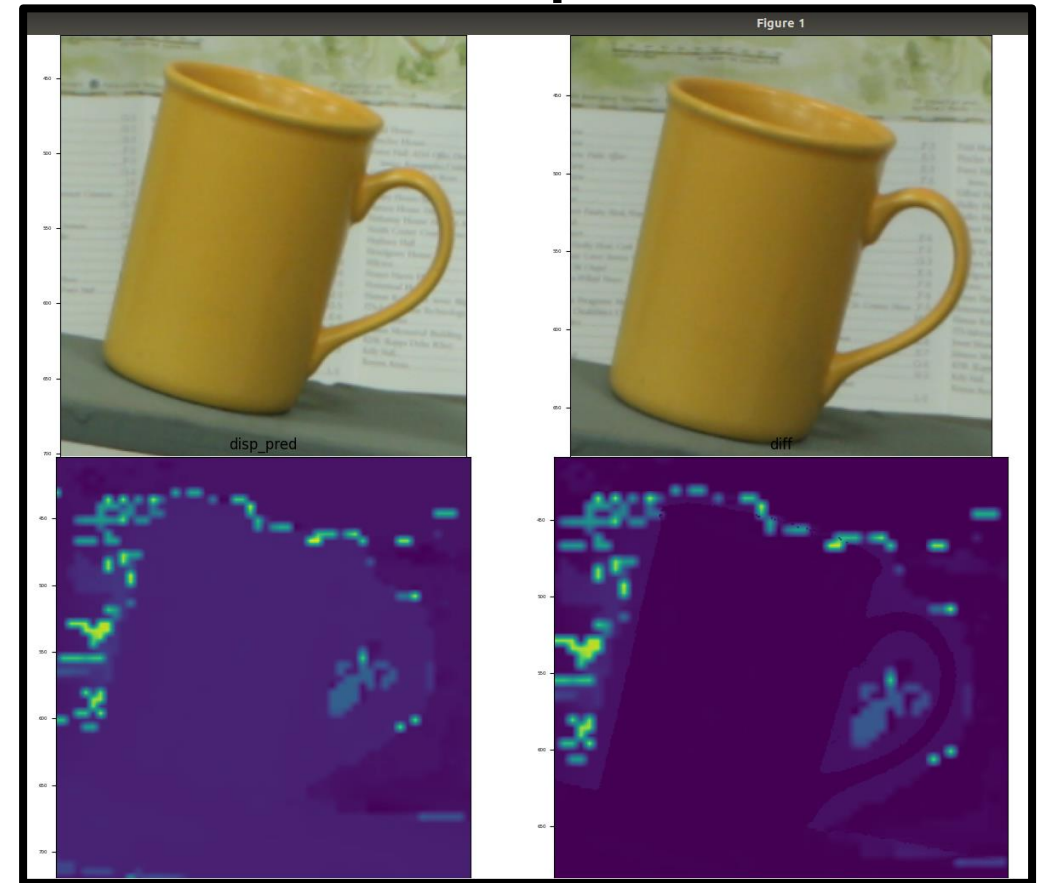
Right image

Difference

# PAC를 stereo matching에 그대로 사용할 때 발생하는 문제점 (scale level=0 output)



PAC



Base model

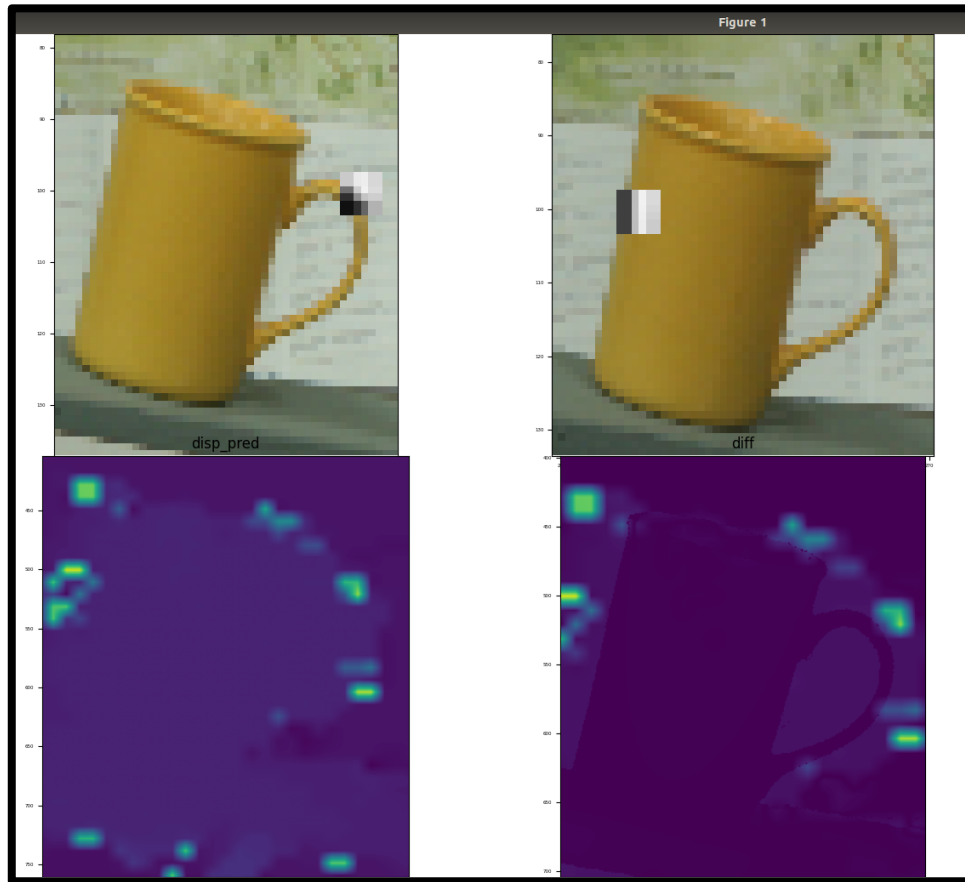
(PAC 모듈을 전부 일반 conv2d로 바꿈)

물체의 경계에서 오차 개선이 이뤄지지 않음

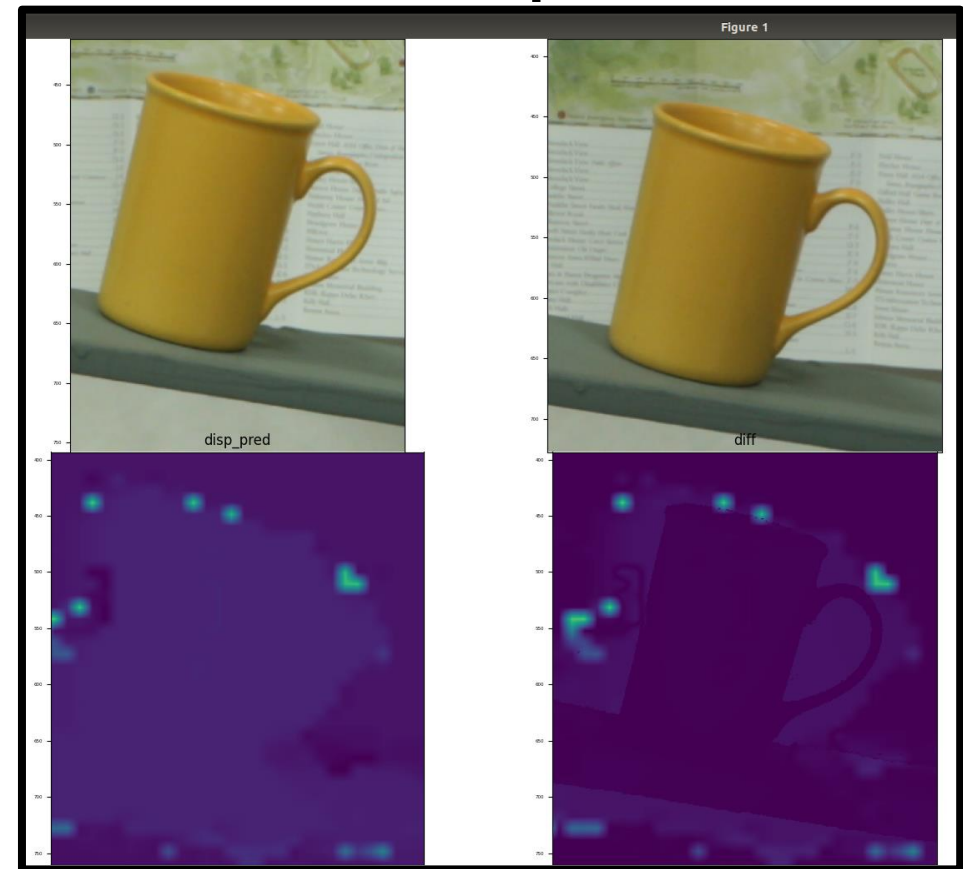
(difference를 보면 여전히 노란 컵 손잡이의 경계를 넘어 디스패리티가 번지고 있음을 알 수 있음)



# PAC를 stereo matching에 그대로 사용할 때 발생하는 문제점 (scale level=1 output)



PAC



Base model

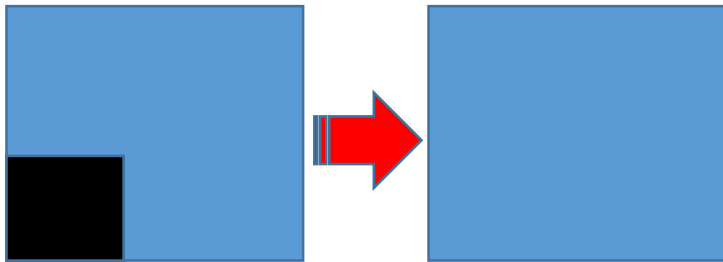
(PAC 모듈을 전부 일반 conv2d로 바꿈)

물체의 경계에서 오차 개선이 이뤄지지 않음

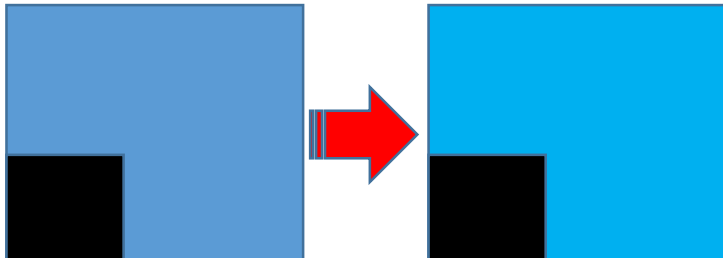
(difference를 보면 여전히 노란 컵 손잡이의 경계를 넘어 디스패리티가 번지고 있음을 알 수 있음)

# PAC를 stereo matching에 사용할 때 발생하는 문제점

- Kernel을 곱한 후 feature 값이 0에 가깝게 되어 원래는 무시되어야 할 픽셀도 feature 값이 완전히 0이 아닌 이상 매칭 시 영향을 미치는 것으로 추정됨.
- 위 추정이 사실이면 물체의 경계에서 오차가 개선되기 힘들.



의도:  
커널을 사용하여 검정 부분(feature=0, !=0)은 매칭에서 영향력을 발휘하지 않도록 제한함.

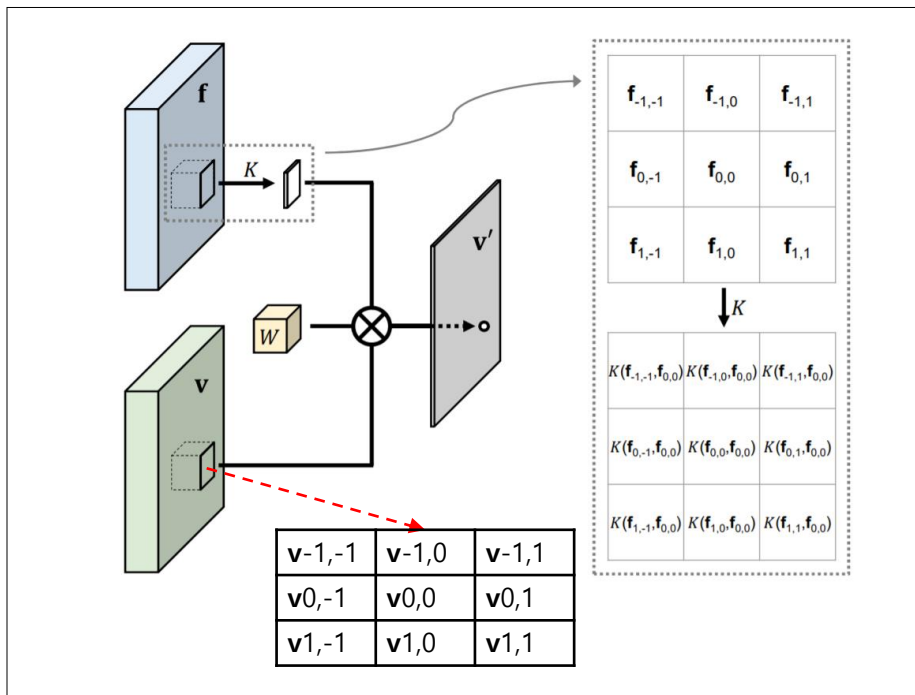


실제:  
검정 부분(feature=0, !=0)이 의도와 다르게 영향력이 있는 것으로 추정됨.

# Kernel 적용 후 보정

- 여전히 영향력을 가지고 있을지도 모르는 feature  $\neq$  0인 부분을 다른 값으로 덮어주고자 함.

# Kernel 적용 후 보정



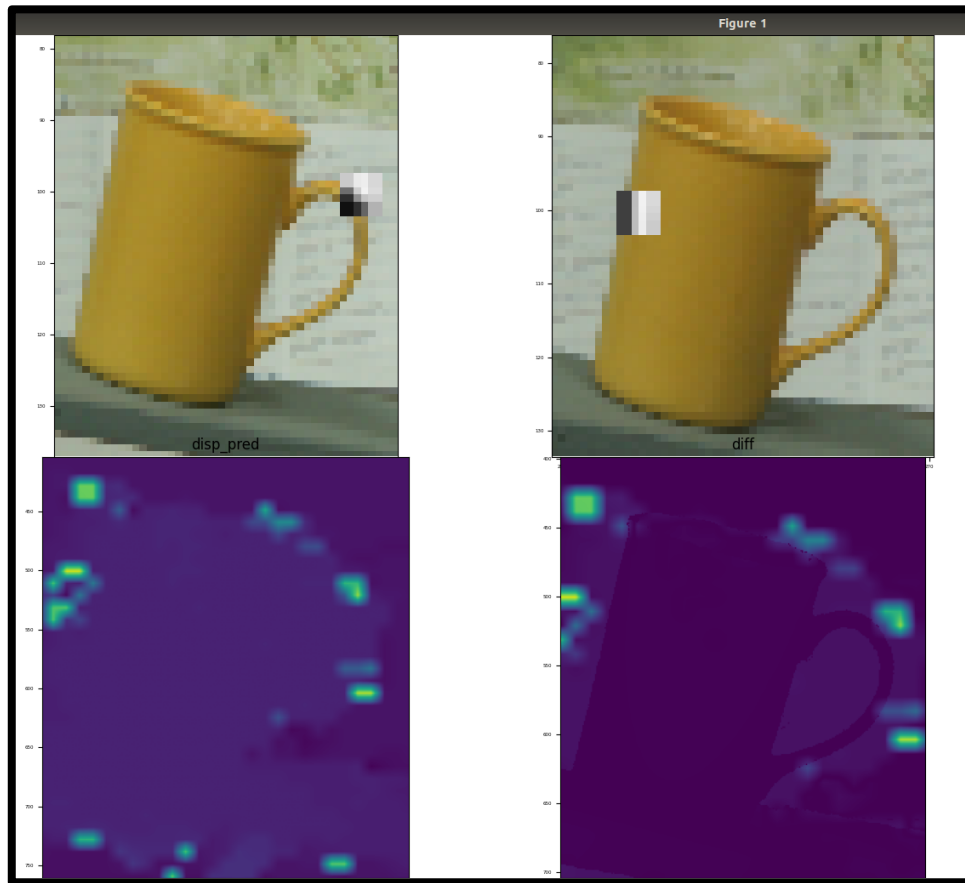
$$wm = \frac{\sum \begin{bmatrix} v_{-1,-1} & v_{-1,0} & v_{-1,1} \\ v_{0,-1} & v_{0,0} & v_{0,1} \\ v_{1,-1} & v_{1,0} & v_{1,1} \end{bmatrix} \otimes \begin{bmatrix} K(f_{-1,-1}, f_{0,0}) & K(f_{-1,0}, f_{0,0}) & K(f_{-1,1}, f_{0,0}) \\ K(f_{0,-1}, f_{0,0}) & K(f_{0,0}, f_{0,0}) & K(f_{0,1}, f_{0,0}) \\ K(f_{1,-1}, f_{0,0}) & K(f_{1,0}, f_{0,0}) & K(f_{1,1}, f_{0,0}) \end{bmatrix}}{\sum \begin{bmatrix} K(f_{-1,-1}, f_{0,0}) & K(f_{-1,0}, f_{0,0}) & K(f_{-1,1}, f_{0,0}) \\ K(f_{0,-1}, f_{0,0}) & K(f_{0,0}, f_{0,0}) & K(f_{0,1}, f_{0,0}) \\ K(f_{1,-1}, f_{0,0}) & K(f_{1,0}, f_{0,0}) & K(f_{1,1}, f_{0,0}) \end{bmatrix}}$$

(weight mean of feature vectors)

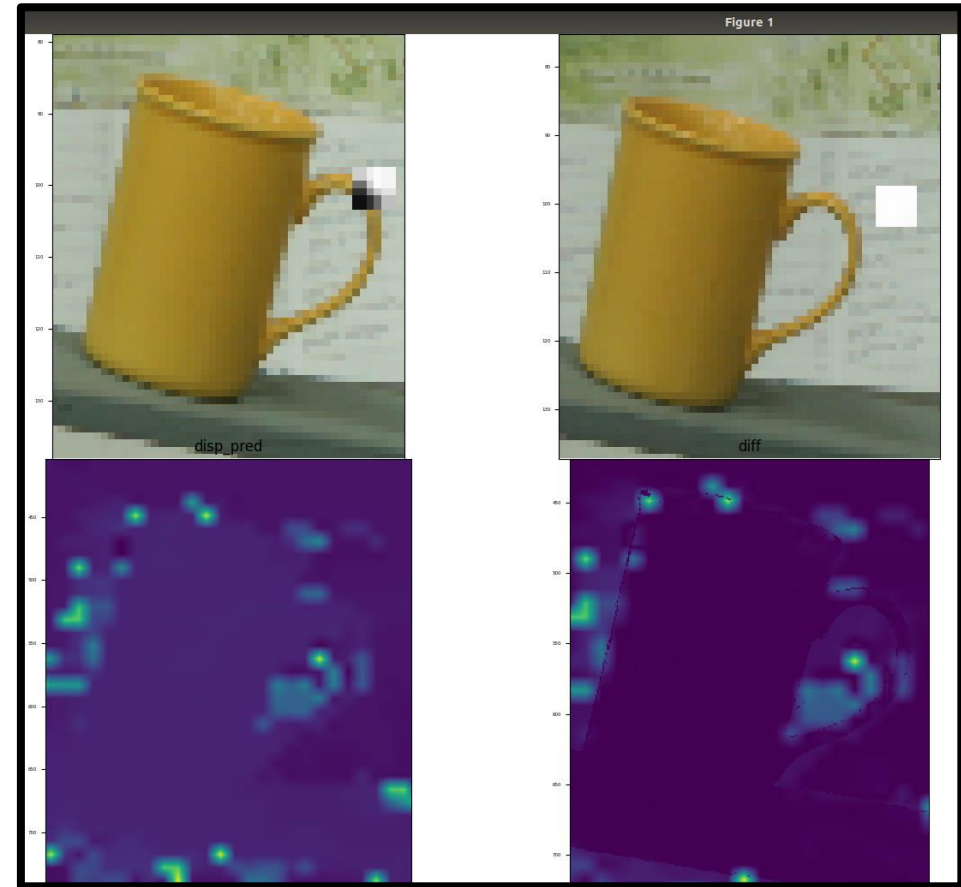
$$\text{corrected\_v\_mul\_kernel} = \begin{bmatrix} v_{-1,-1} & v_{-1,0} & v_{-1,1} \\ v_{0,-1} & v_{0,0} & v_{0,1} \\ v_{1,-1} & v_{1,0} & v_{1,1} \end{bmatrix} \otimes \begin{bmatrix} K(f_{-1,-1}, f_{0,0}) & K(f_{-1,0}, f_{0,0}) & K(f_{-1,1}, f_{0,0}) \\ K(f_{0,-1}, f_{0,0}) & K(f_{0,0}, f_{0,0}) & K(f_{0,1}, f_{0,0}) \\ K(f_{1,-1}, f_{0,0}) & K(f_{1,0}, f_{0,0}) & K(f_{1,1}, f_{0,0}) \end{bmatrix} + \begin{bmatrix} wm & wm & wm \\ wm & wm & wm \\ wm & wm & wm \end{bmatrix} \otimes (1 - \begin{bmatrix} K(f_{-1,-1}, f_{0,0}) & K(f_{-1,0}, f_{0,0}) & K(f_{-1,1}, f_{0,0}) \\ K(f_{0,-1}, f_{0,0}) & K(f_{0,0}, f_{0,0}) & K(f_{0,1}, f_{0,0}) \\ K(f_{1,-1}, f_{0,0}) & K(f_{1,0}, f_{0,0}) & K(f_{1,1}, f_{0,0}) \end{bmatrix})$$

$$v' = \text{Relu}(\text{corrected\_v\_mul\_kernel} \otimes W + \text{bias})$$

# 보정 전 vs 보정 후 (scale level=1 output)



PAC  
(before)



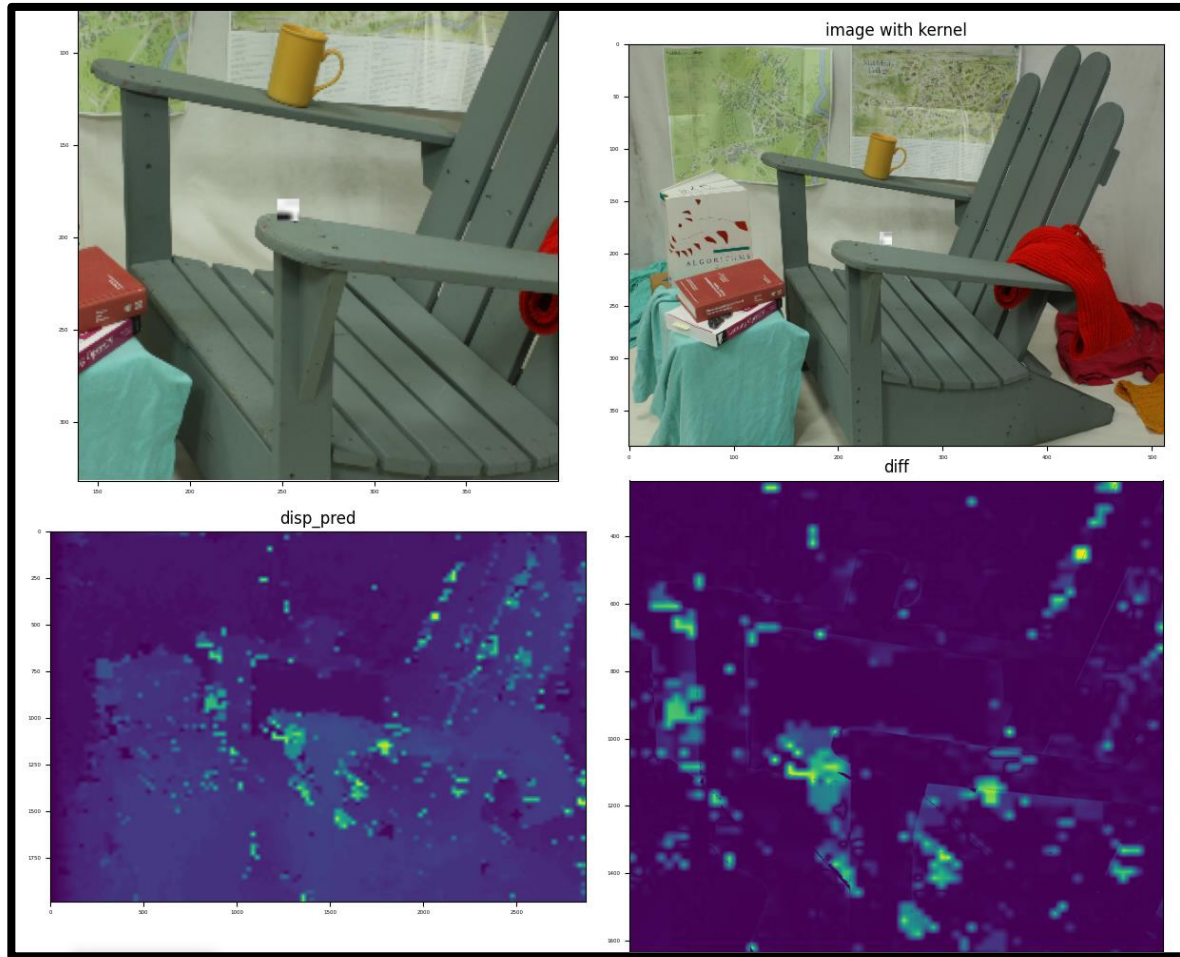
PAC  
(after)

물체의 경계에서 디스패리티가 번지는 현상이 개선됨.

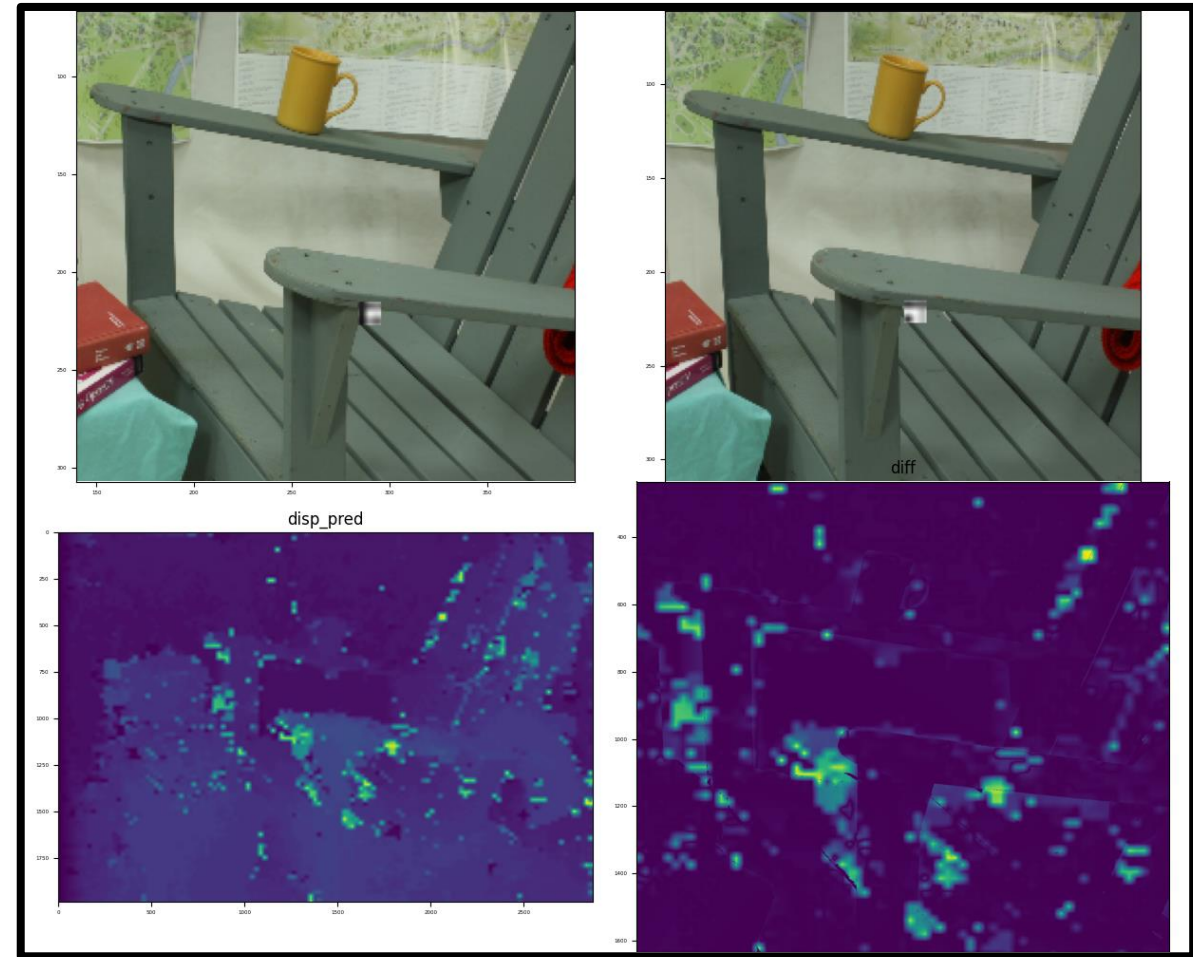
# 문제점

- 무시되어야 할 픽셀의 커널(=어텐션?) 값이 0에 가깝지 않으면 보정 효과가 감소함.
- 무시되어야 할 픽셀의 커널 값이 0에 가깝더라도 완전히 0이 아닌 이상, 커널 값이 높은 픽셀의 텍스처가 너무 단조로우면 무시되어야 할 픽셀의 영향을 받을 수 있음
- => 가이드를 정말 잘 만들어야 함...

# 문제점 (scale\_level=2 output)



커널 검정색 부분의 값이 약 0.13,  
의자 팔걸이 상단부에서는 디스패리티가 번지지 않음

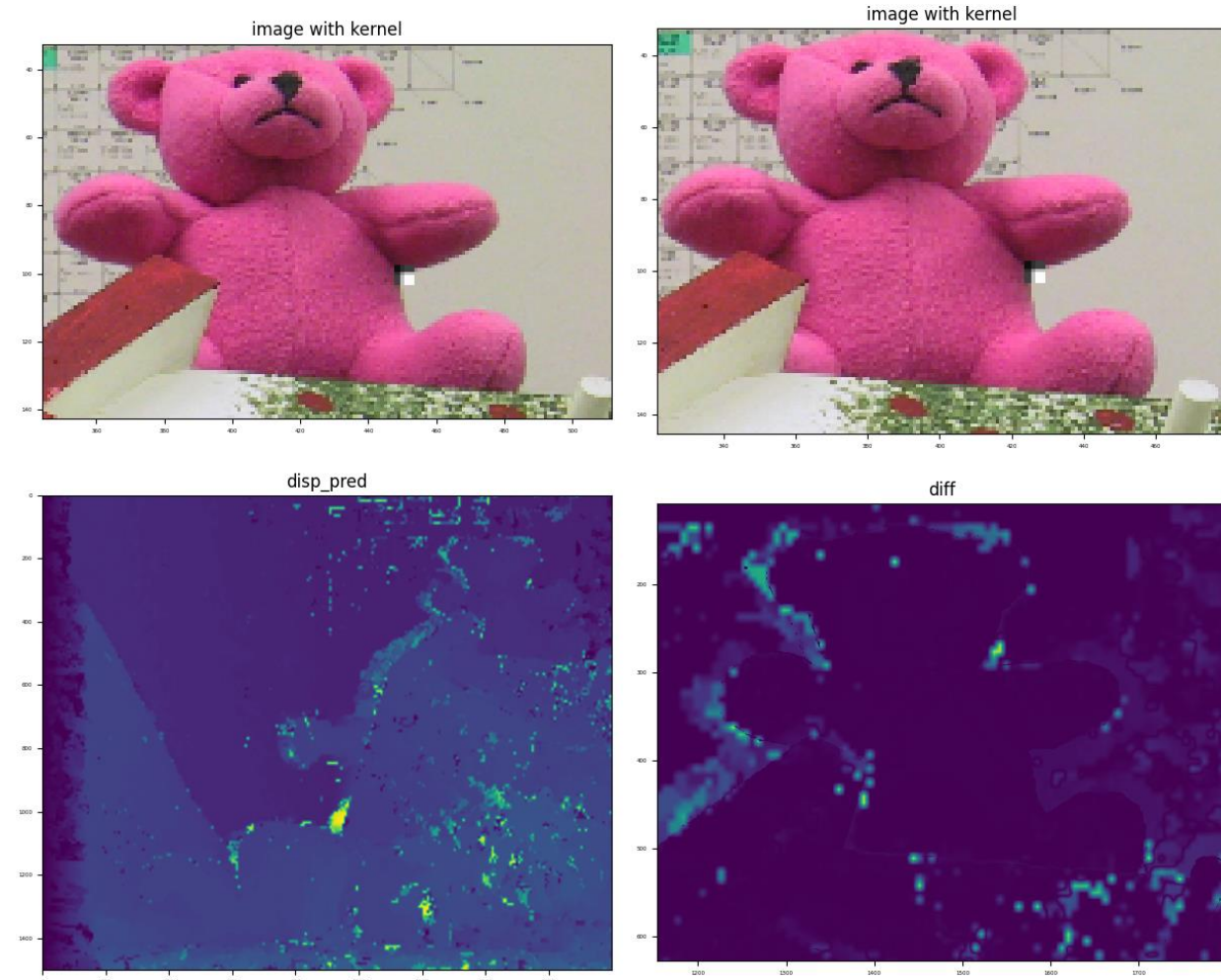


커널 상단 검정색 부분의 값이 각각 약 0.48, 0.37  
의자 팔걸이 하단부에서는 디스패리티가 번짐

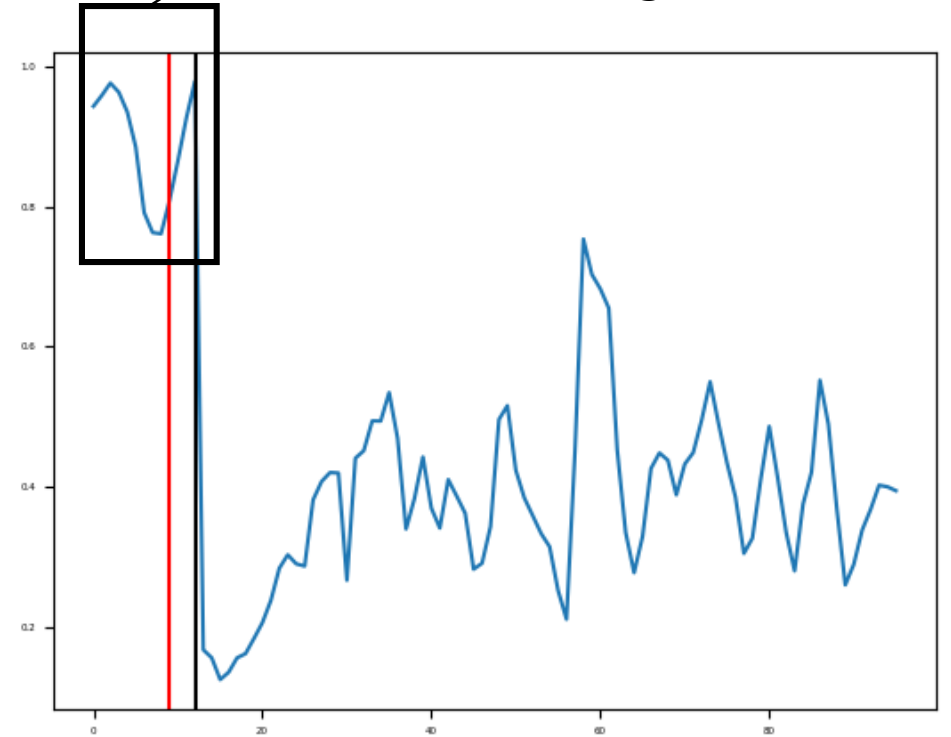


# 문제점 (scale\_level=1 output)

텍스처가 단조로운 영역.  
검은 세로선은 pred  
붉은 세로선은 gt



커널 검정색 부분의 값이 약 0.24임에도  
디스패리티가 번짐.



텍스처가 단조로운 구간에서 매칭 스코어가  
전체적으로 높게 나타나다 보니 약간의 잡음도  
영향을 크게 주는 것으로 생각됨.<sup>16</sup>



# PAC 모듈의 가이드 문제

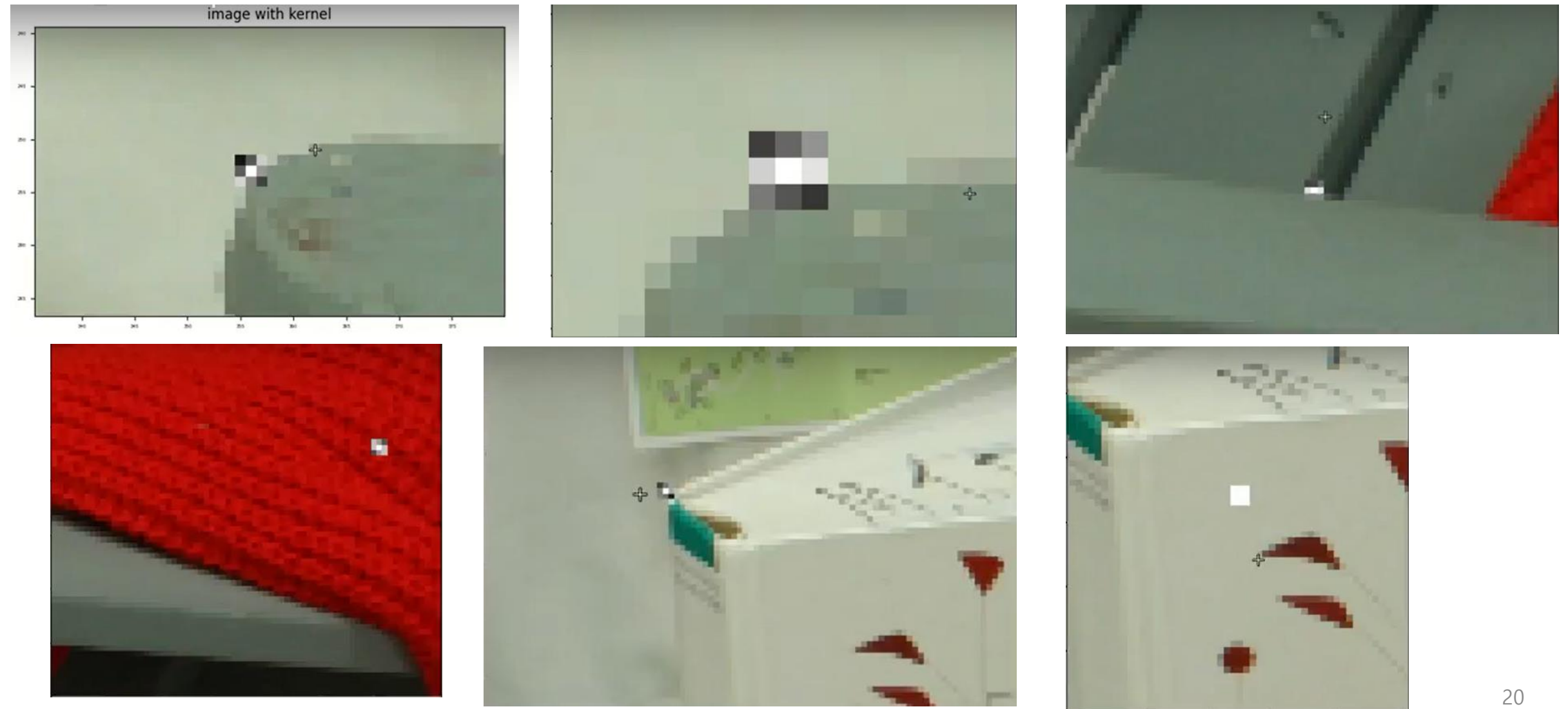
# Guide

- RGB channel:
  - 단점: 자잘한 텍스처에 취약함.
- Output of conv2\_2(or something):
  - Rgb 이미지를 몇 번 컨볼루션 한 후 결과의 feature로 거리를 계산.
  - 단점: 엣지에 크게 영향을 받아서 down scale로 엣지 영향을 줄이지 않으면 사용할 수 없음 (실제 논문도 down scale(bilinear) 하고 사용)
- Sub-network (encoder-decoder, segmentation):
  - RGB 이미지를 인코더-디코더 네트워크로 처리한 결과의 feature로 거리를 계산
  - 단점: 세그멘테이션 전용의 훈련 데이터 셋을 확보하여 따로 훈련하지 않는 이상 Output of conv2\_2의 단점이 그대로 나타나므로 down scale하여 사용해야 함.

# Guide – RGB channel

- PAC 논문에서 joint upsampling task에 사용된 방식
- 다운 스케일이 필요하지 않다는 장점
- 자잘한 텍스처가 많은 영역에서 역효과를 냄.
  - 바이리터럴 필터(가우시안 스무딩)를 먼저 사용해서 자잘한 텍스처를 없애준 후 가이드로 사용 시 텍스처의 영향을 적게 받음
    - 너무 조잡한 방법 같음.
  - 현재 auto encoder 구조를 활용한 텍스처 소거 방안을 생각 중

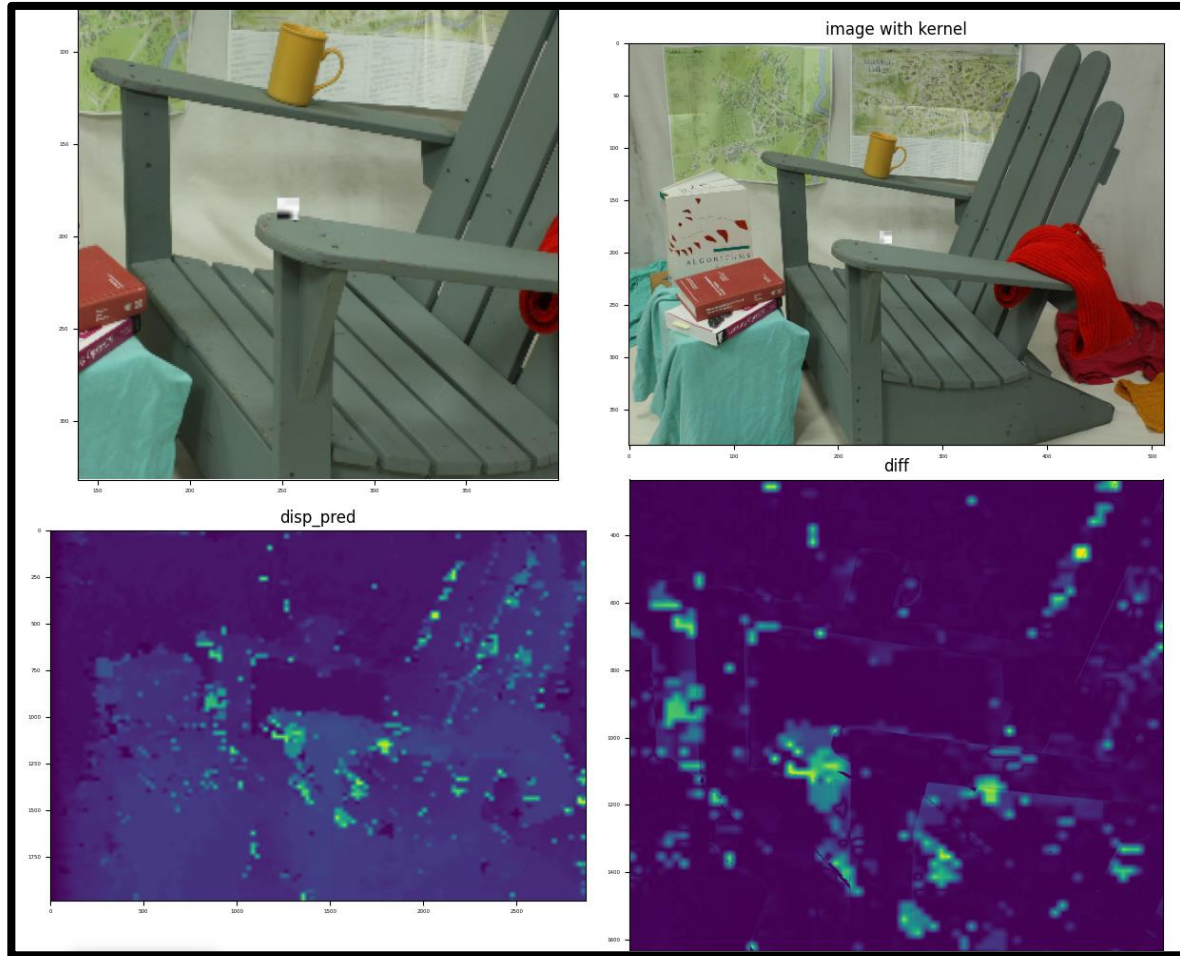
# Guide – Output of conv2\_2



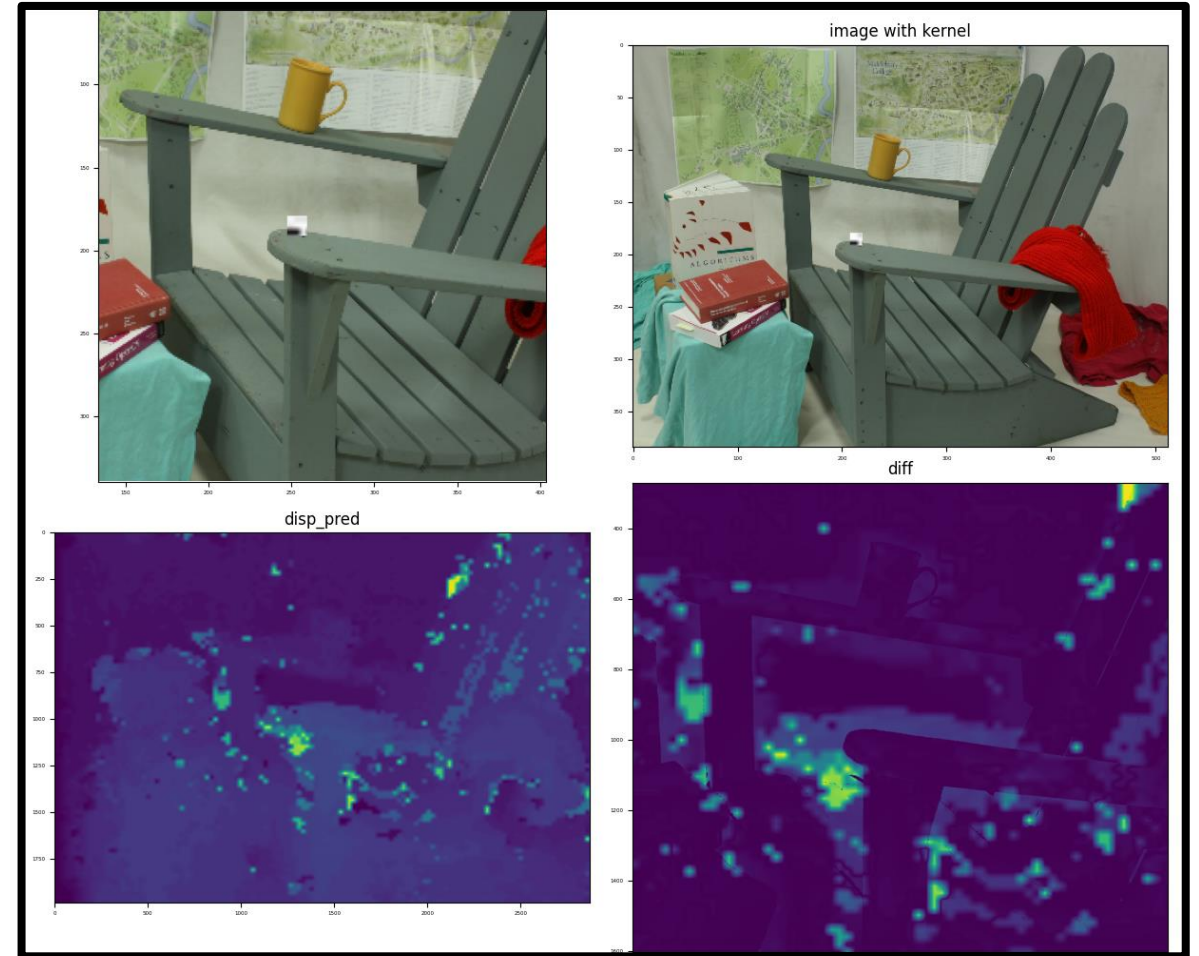
# Guide – Output of conv2\_2

- PAC 논문에서 semantic segmentation task에 사용된 방식
- [https://drive.google.com/drive/u/0/folders/1tyM8tKgtlBHugx6sfziK\\_vQxSnIF6VsV](https://drive.google.com/drive/u/0/folders/1tyM8tKgtlBHugx6sfziK_vQxSnIF6VsV)
  - 동영상 참조
- 다운스케일 하지 않으면 사용할 수 없는 수준.

# 다운스케일이 필요한 가이드 (scale\_level=2 output)



스케일 레벨 0, 1, 2에 전부 가이드 적용



스케일 레벨 0, 1은 가이드를 적용하지 않고  
스케일 레벨 2에만 가이드 적용한 경우<sup>22</sup>

# 다운스케일이 필요한 가이드

- 다운스케일이 필요한 가이드의 경우 scale level=0~1인 계층에서는 가이드를 이용하지 못하게 되는데 상위 레벨에서 이미 다른 물체의 feature가 섞여 들어가버리기 때문에 하위 레벨에서 디스패리티가 번지는데 영향을 줄 수 있다고 생각됨.
- 따라서 다운스케일이 필요한 가이드 생성 방법은 고려하지 않겠음.

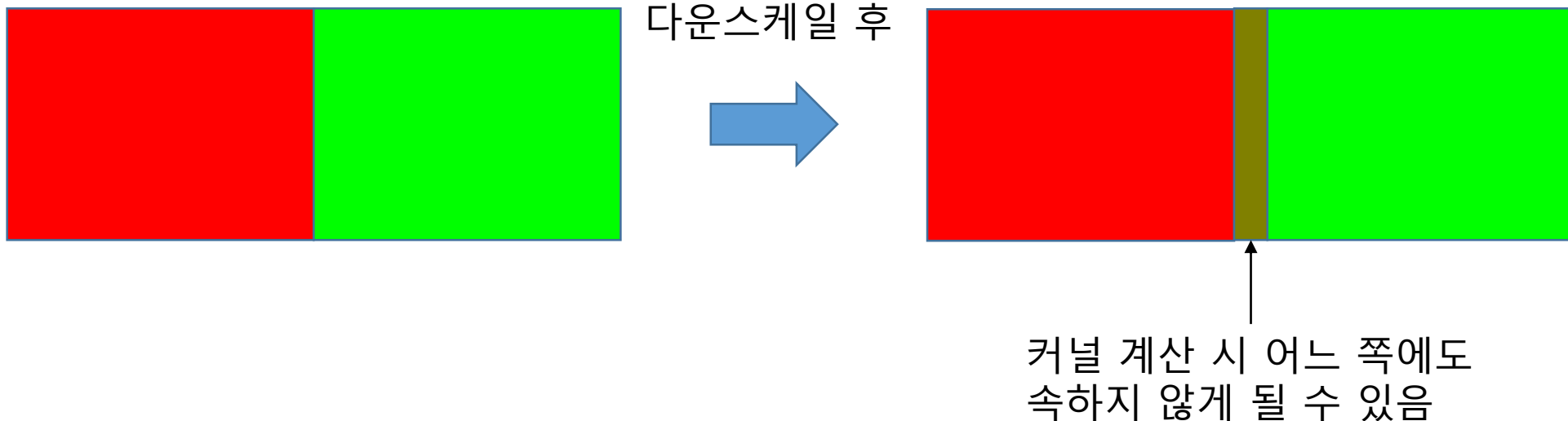
# Guide의 다운 스케일 방법

- PAC 논문 저자의 모델과 지금 사용 중인 모델에서는 스케일 별로 가이드를 요구하기 때문에 저자의 방식대로 하나의 가이드를 다운 스케일하여 사용하고자 함.
- Nearest 방식 혹은 PacPool 모듈을 쓰는 것이 더 나아보이며, 현재는 PacPool 모듈에 Gaussian 스무딩 기능을 추가하여 사용하는 중.



# Guide의 다운 스케일 방법

- Bilinear 방식을 사용할 경우 물체의 경계에서 RGB값이 섞이기 때문에 주변 물체의 픽셀 값에 대한 경계의 픽셀 값의 1/2 거리가 동시에 멀어질 수 있음



# 향후

- Multiscale attention
  - PAC joint upsampling을 사용한 scale level의 통합?
- Confidence와 변화율 사용
  - Confidence 값이 일정 수치 아래면 주변 픽셀의 디스패리티 변화율로부터 디스패리티 계산