



RoomNet, LayoutNet, HorizonNet, HoHoNet

2021. 09. 07

AI융합학부 길다영

CONTENTS

1

RoomNet

<https://github.com/GitBoSun/roomnet>

2

LayoutNet

<https://github.com/sunset1995/pytorch-layoutnet>

3

HorizonNet

<https://github.com/sunset1995/HorizonNet>

4

HoHoNet

<https://github.com/sunset1995/HoHoNet>

5

정리

https://github.com/arittung/3D_Room_Reconstruction





1. RoomNet



2. LayoutNet

2 LayoutNet



LayoutNet

- 파노라마와 투시 이미지, 직육면체 레이아웃 및 보다 일반적인 레이아웃(예: "L"자형 룸)에 걸쳐 일반화하는 단일 이미지에서 룸 레이아웃을 예측하는 알고리즘
 - 단일 RGB 등각 파노라마에서 직접 레이아웃을 추정하여 reconstruction을 단순화함.
 - 최종 출력물은 카메라와의 각 벽의 거리, 높이 및 레이아웃 회전으로 매개 변수화된 희박하고 콤팩트한 평면 Manhattan layout 이다.
-
- 소실점을 기준으로 파노라마 이미지를 정렬한 후, 시스템은 심층 네트워크를 사용하여 파노라마 이미지의 경계와 모서리를 직접 예측한다. 이러한 점에서 우리는 심층 네트워크를 사용하여 투시 이미지의 레이아웃 모서리와 가시적인 모서리를 나타내는 레이블을 직접 예측하는 RoomNet과 유사함.
-
- ① 단일 파노라마에서 3D 직육면체 레이아웃을 예측한다.
 - ② 단일 파노라마에서 3D non-cuboid Manhattan layout을 추정한다.
 - ③ 단일 원근 이미지에서 레이아웃을 추정한다.

2 LayoutNet



내용 정리

- ① “Manhattan World” 가정 하에 3D 모델을 만들기 위해 수평 카메라가 촬영한 투시 이미지에 바닥/벽 경계를 맞춘다
- ② 방향 지도(Orientation Maps)를 작성하고, 검출된 선분을 기반으로 레이아웃 가설을 생성하고, 그 중에서 가장 적합한 레이아웃을 선택한다.
- ③ 소실점 3개를 풀고, 소실점과 일치하는 레이아웃을 샘플링하고, 가장자리와 기하학적 맥락 일관성을 기반으로 최적의 레이아웃을 선택하여 입체 레이아웃을 복구한다.
→ 소실점을 기준으로 파노라마 이미지를 정렬한 후, 시스템은 심층 네트워크를 사용하여 파노라마 이미지의 경계와 모서리를 직접 예측한다.



RoomNet에 대한 개선점

- 소실점을 기반으로 이미지를 정렬하고, 다중 레이아웃 요소(모서리, 경계, 크기 및 변환)를 예측하고, 결과 예측에 제한된 맨해튼 레이아웃을 맞추기 때문에 개선을 보여준다.
- 파노라마 영상에 적용함.
- 정렬단계와 경계, 모서리 및 3D cuboid parameter에 대한 다중 작업 예측에서 다름.
- RoomNet은 RNN을 사용하여 2D 모서리 위치 예측을 세분화하지만 이러한 예측은 3D 입체 배치와 일치하지 않을 수 있다.



궁금한 점.

- ① ?

2 LayoutNet - 소개

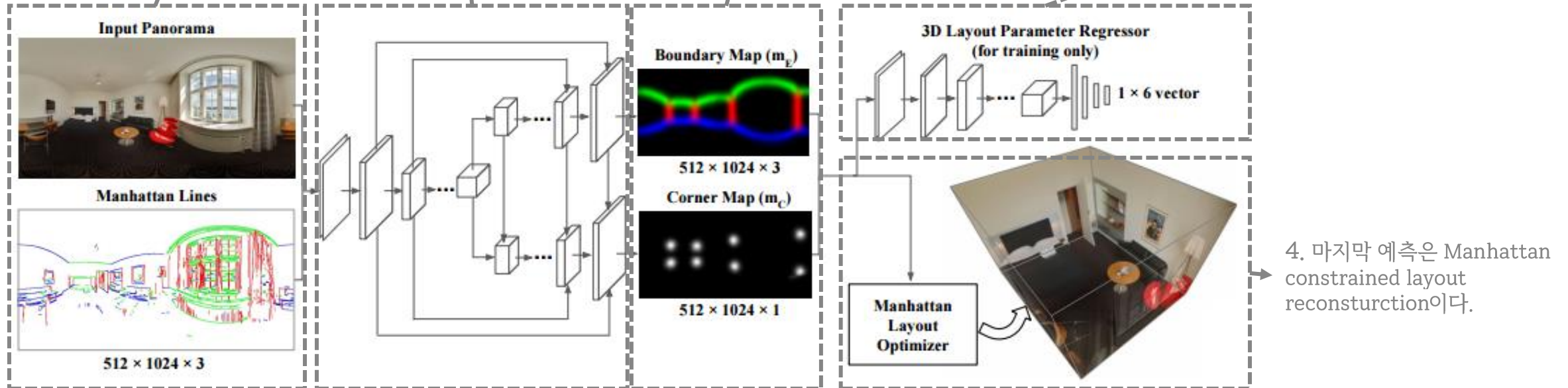
LayoutNet은 encoder-decoder 전략을 따른다.

1. 네트워크 입력은 단일 RGB 파노라마와 Manhattan line map의 결합이다.

2. 네트워크는 레이아웃 경계와 모서리 위치를 공동으로 예측한다.

3. 3D 레이아웃 파라미터 손실은 정확도를 최대화하는 예측을 장려한다.

네트워크
: encoder-decoder 전략

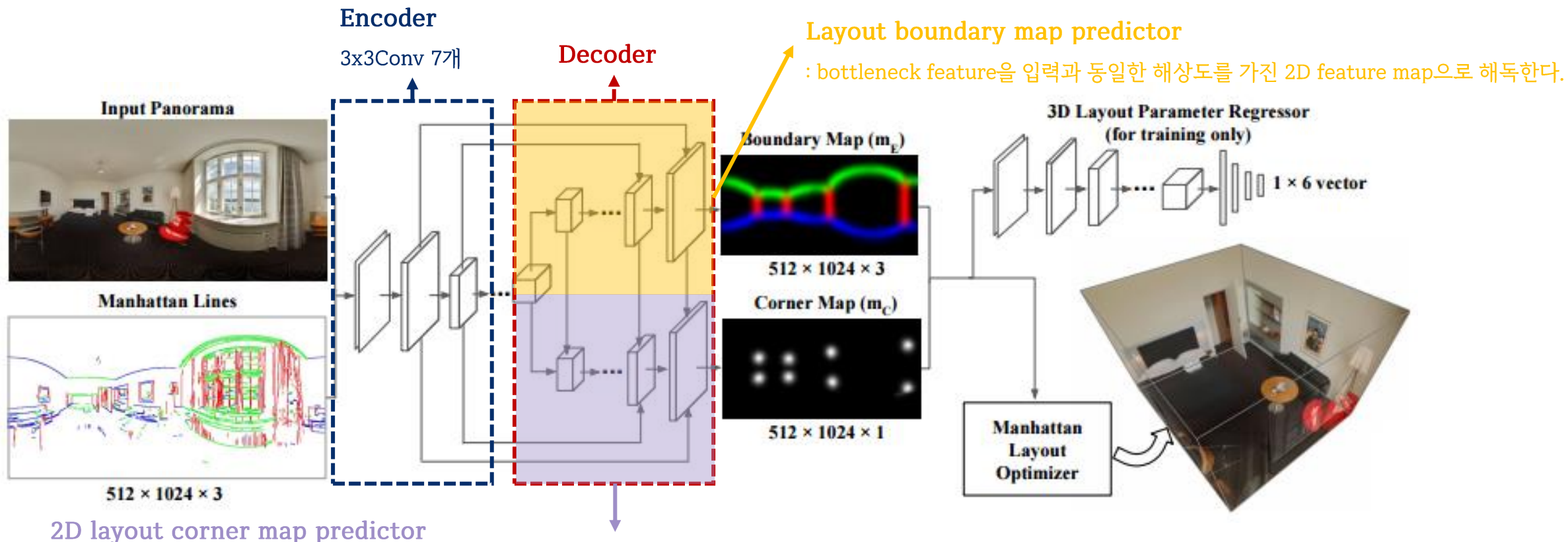


- ① 시스템이 소실점을 분석하고 이미지를 바닥과 수평으로 정렬한다. 이러한 정렬을 통해 벽면 경계가 수직선임을 보장하고 실험에 따라 오류를 줄일 수 있다.
- ② 인코더-디코더 구조와 skip connections을 가진 CNN을 사용하여 이미지에서 직접 코너(레이아웃 접합)와 경계 확률 맵(corner (layout junctions) and boundary probability maps)을 예측한다. 모서리와 경계는 각각 방 배치를 완벽하게 표현해 준다.
- ③ 3D Layout parameters는 예측된 모서리와 경계에 적합하도록 최적화된다

2

LayoutNet - ① 파노라마 이미지 정렬 + ② corner (layout junctions) and boundary probability maps 예측

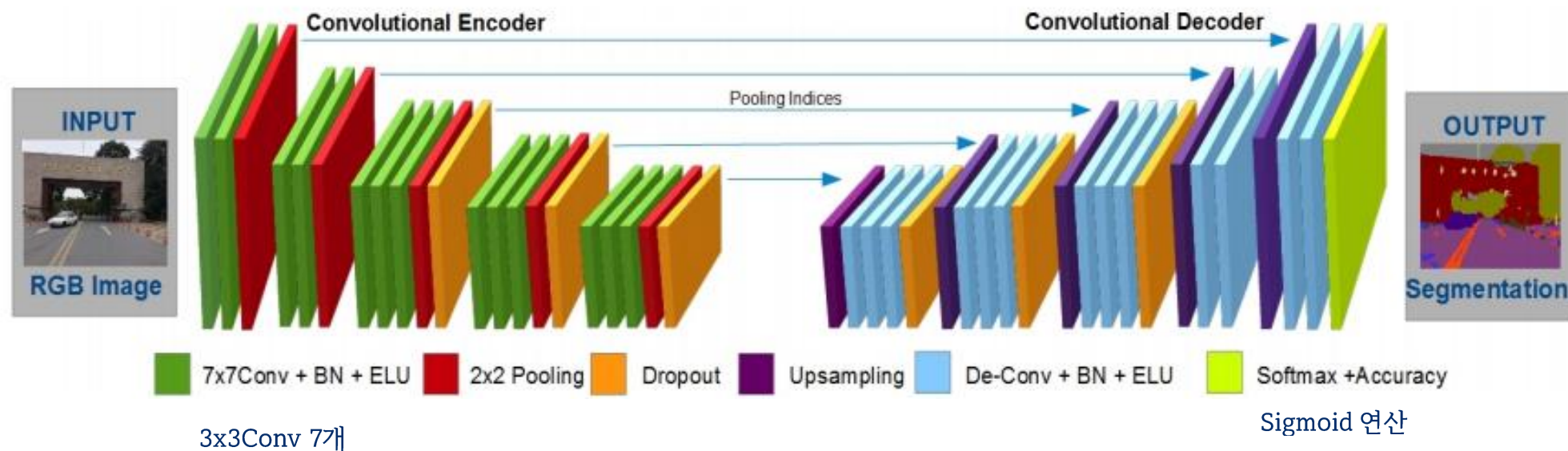
- 구형 투영에서 바닥 평면 방향을 추정하여 이미지를 정렬하고 장면을 회전한 다음 2D 등각 투영으로 다시 투영한다
we first align the image by estimating the floor plane direction under spherical projection, rotate the scene, and reproject it to the 2D equirectangular projection.
- 네트워크는 encoder-decoder 전략을 따른다.
- 훈련 방법 : 2D 레이아웃 예측 네트워크의 경우, 먼저 네트워크의 매개 변수를 초기화하기 위해 레이아웃 경계 예측 작업을 훈련한다.



2

LayoutNet - ① 파노라마 이미지 정렬 + ② corner (layout junctions) and boundary probability maps 예측

An Encoder-Decoder Based Convolution Neural Network (CNN)

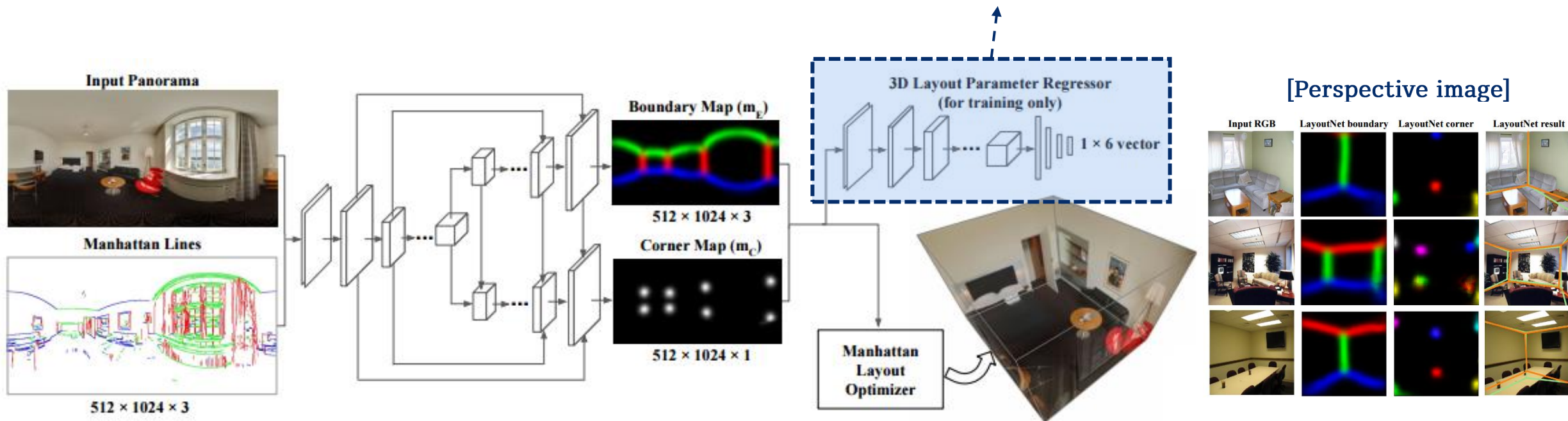


2 LayoutNet – ③ 3D layout Parameter Regressor

- 3D 레이아웃 파라미터에 대한 회귀 분석기를 만드는 것보다는 더 나은 모서리와 경계를 만들기 위해 회귀 분석기를 훈련한다.
- 회귀 분석기는 2D 영상에 다시 투영할 수 있는 3D 레이아웃의 파라미터를 출력하여 end to end 예측 접근 방식을 제시한다.
- 3D 회귀기는 두 예측된 2D map의 연결을 입력으로 얻고 3D 레이아웃의 매개변수를 예측한다.
- 3D 회귀 분석기가 정확하지 않다는 것이 관찰된다. Loss object에 3D 회귀 분석기를 포함하면 네트워크의 예측이 약간 개선되는 경향이 있다.
- 훈련 방법 : 3D 레이아웃 회귀기의 경우, 먼저 지상 실측 레이아웃 경계와 모서리를 입력으로 하여 네트워크를 훈련한 다음 이를 2D 레이아웃 decoder와 연결하고 전체 네트워크 end-to-end를 훈련한다.

Non-cuboid 레이아웃에 대해서는 3D parameter regressor로 훈련하지 않는다.

Perspective image예측할 때, 이미지의 모서리와 경계를 직접 예측하는 대신 정렬 및 최적화 단계를 건너뛰고, 또한 regressor branch를 사용하지 않는다.



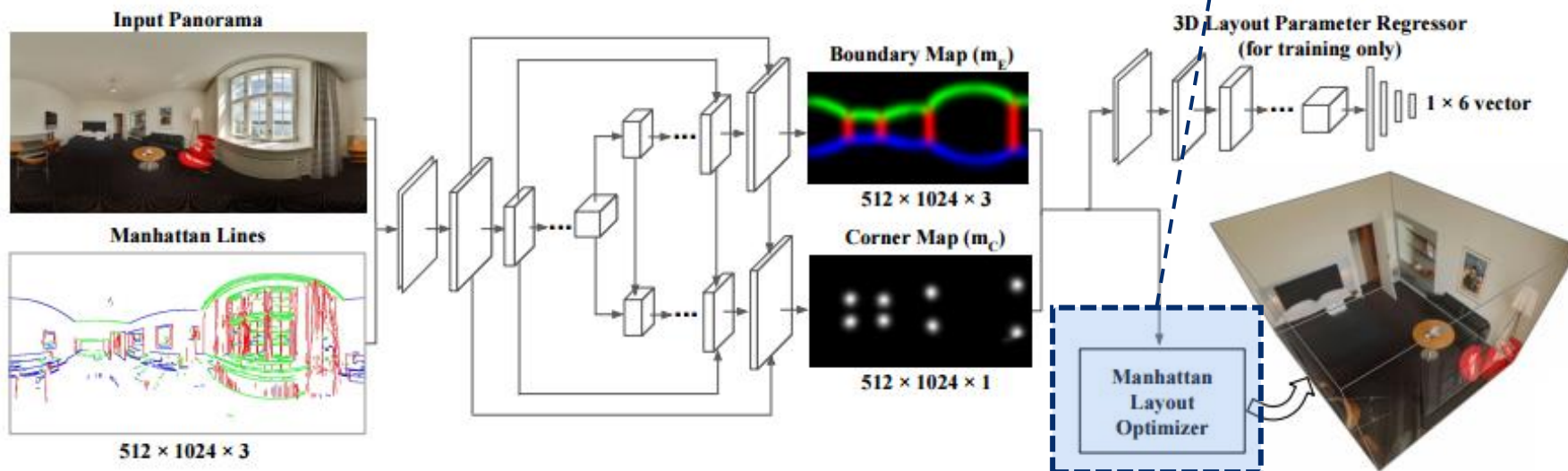
2 LayoutNet – ④ 3D layout Optimazation

- 초기 2D 코너 예측은 네트워크가 출력하는 corner probability maps에서 얻어진다.

- ① 먼저, 각 열에 대한 반응 합계를 얻기 위해 여러 행에 걸쳐 반응을 합산한다.
- ② 로컬 최대값은 로컬 최대값 사이의 거리가 최소 20픽셀인 column responses에서 발견된다.
- ③ 선택한 열을 따라 가장 큰 두 개의 봉우리가 표시된다.

이러한 2D 코너는 Manhattan 제약 조건을 충족하지 못할 수 있으므로, 우리는 추정치를 구체화하기 위해 최적화를 수행한다.

- 예측된 모서리 위치가 주어지면, 하단 모서리가 동일한 지면 위에 있고 상단 모서리가 하단 모서리 바로 위에 있다고 가정하여 카메라 위치와 3D 레이아웃을 최대 규모 및 변환까지 직접 복구할 수 있다.
- 또한 배치 형태를 Manhattan으로 제한하여 교차 벽이 수직이 되도록 할 수 있다.





3. HorizonNet

3 HorizonNet



HorizonNet

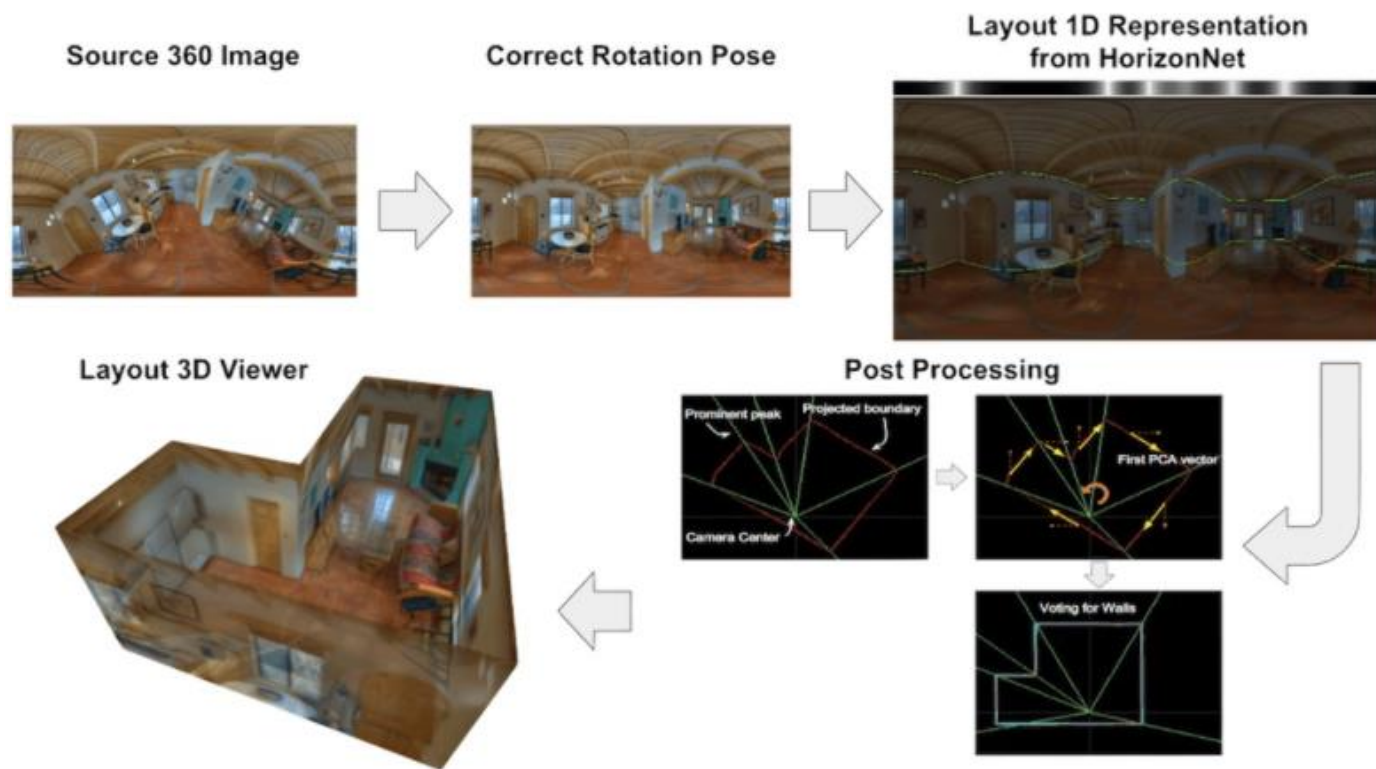
목표 : 360도 파노라마 이미지로부터 Manhattan room layout 추정



궁금한 점.

- ① Pano Stretch Data Argumentaion는 어디에서 쓰이는가?
- ② 추가 논문해석이 필요해보임.

3 HorizonNet - 소개



1. 수직 보정 전처리를 통해 1D Layout 표현
2. 특징 추출
3. Post-Processing

3

HorizonNet - ① 1D Layout Representation + ② 특징 추출(feature extractor)

- 입력으로 받은 파노라마 사진을 수직 보정하여 소실점과 edge를 찾는다.
- ResNet50과 LSTM을 이용하여 훈련된 모델로 사진의 특징을 추출하여 천장-벽 경계, 바닥-벽 경계, 벽-벽 경계가 표시된 1D Layout을 도출한다.

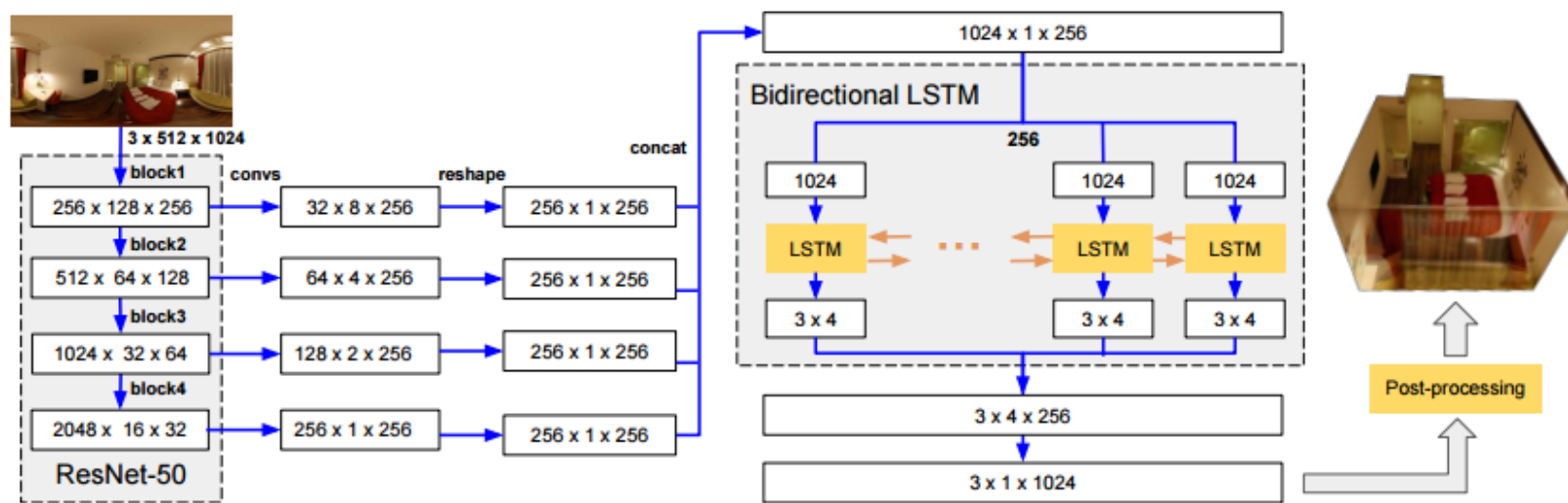


Figure 2: An illustration of the HorizonNet architecture.

[1D Layout]



Figure 3: Visualization of our 1D ground truth representation.

y_c : 천장-벽 경계

y_f : 바닥-벽 경계

y_w : 벽-벽 경계

3 HorizonNet - ③ Post-processing

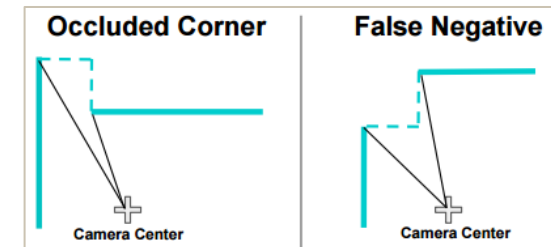
- Manhattan World 가정으로 바닥 천장, 벽면을 복구.
- 공식을 통해 천장-바닥 거리를 계산한 후 벽면을 복구함.

1. 빨간 선 : 구해진 천장-벽의 경계
녹색 선 : 눈에 띄는 모서리를 여러 부분으로 나눈 것

2. 각 벽면에서 PCA 벡터를 얻고, 모든 벡터의 평균화된 각도로 전체 평면을 회전시킴.

3. 모든 벽면에 대해서 XZ축과 수직인 벽에 투표를 하여 투표에서 가장 많은 표를 얻은 벽이 선택됨.

투표를 할 때는 Cuboid 형태의 방과 Non-Cuboid 형태의 방으로 나뉘어 진행됨. → 구조에 따른 벽면의 개수에 차이가 존재하고, Non-Cuboid의 구조에서 두가지의 특이점이 존재하기 때문

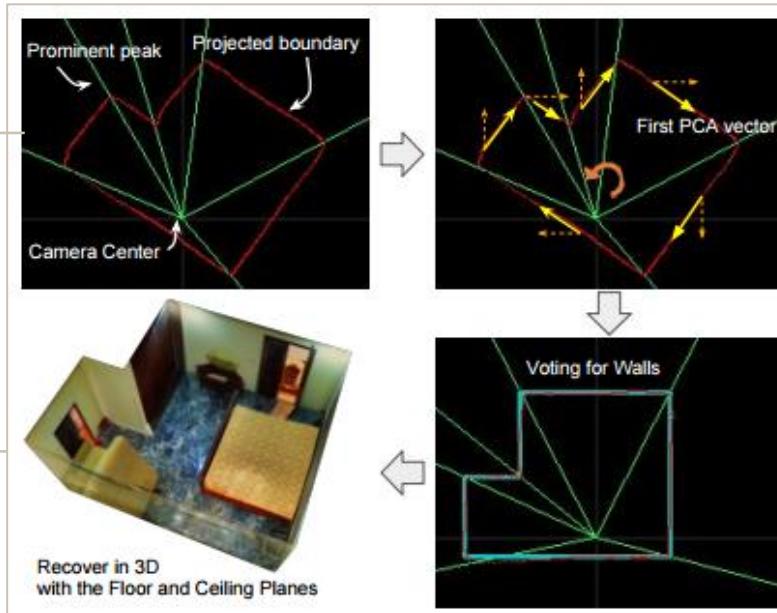


『Non-Cuboid 구조에서 두가지 특이점』

- I. 가려져서 보이지 않는 코너가 존재하는 경우
- II. 코너가 존재하지 않고 훈련된 모델이 코너가 없다고 판단한 경우.

이 두 경우에는 기존과 같이 벽에 투표하지 않고 두 개의 두드러진 모서리와 두 개의 벽의 위치에 따라 코너를 추가하는 특별한 방식을 사용한다.

4. 투표하여 결정된 천장, 바닥, 벽을 기준으로 3D Room이 복원된다.



Manhattan World 가정 : 영상에 나타나는 평면들은 3차원상에서 서로 직교하는 평면들 로만 이루어져 있다.

Cuboid : 직육면체의 형태

PCA : 분산이 가장 큰 방향이 가장 주된 방향이라고 가정, 다차원의 데이터 분포를 가장 잘 표현하는 성분들을 찾아 주기 위함



4. HoHoNet

4 HoHoNet



HoHoNet

높이 치수가 평평한 잠재 수평 특징 표현(LHFeat)을 통해 레이아웃 구조, 조밀한 깊이 및 의미 분할을 모델링하기 위한 새로운 딥러닝 프레임워크.



HoHoNet은 두 가지 측면에서 발전했다

- ① 심층 아키텍처는 향상된 정확도로 더 빠르게 실행되도록 재설계됨.
- ② LHFeat의 픽셀당 조밀한 예측을 가능하게 하여 열당 출력 형태 제약을 완화하는 새로운 수평선 대 밀도 모듈을 제안함.



궁금한 점.

- ① 열당 예측과 픽셀당 예측의 차이?

⇒ 열 당 예측으로 layout 을, 픽셀당 예측으로 depth와 semantic을 나오게 하는 것 같다.

- ① Hohonet은 Depth estimation과 semantic segmentation이 나오고 그걸 horizonnet처럼 3d room reconstruction을 하는 듯..? → 이걸 코드에서 알 수 있을까?

⇒ 알 수 있다!! `HoHoNet/lib/model/modality/`에 각 코드가 나와 있음.

- ② 실행시 room layout이 만들어지지만, 코드 상 depth와 semantic 을 사용한 부분, 그리고 그게 결과로 나타난 부분이 있는지 알아봐야 할 거 같다

4 HoHoNet

! 내용 정리

Hohonet의 결과 : layout, depth, semantic

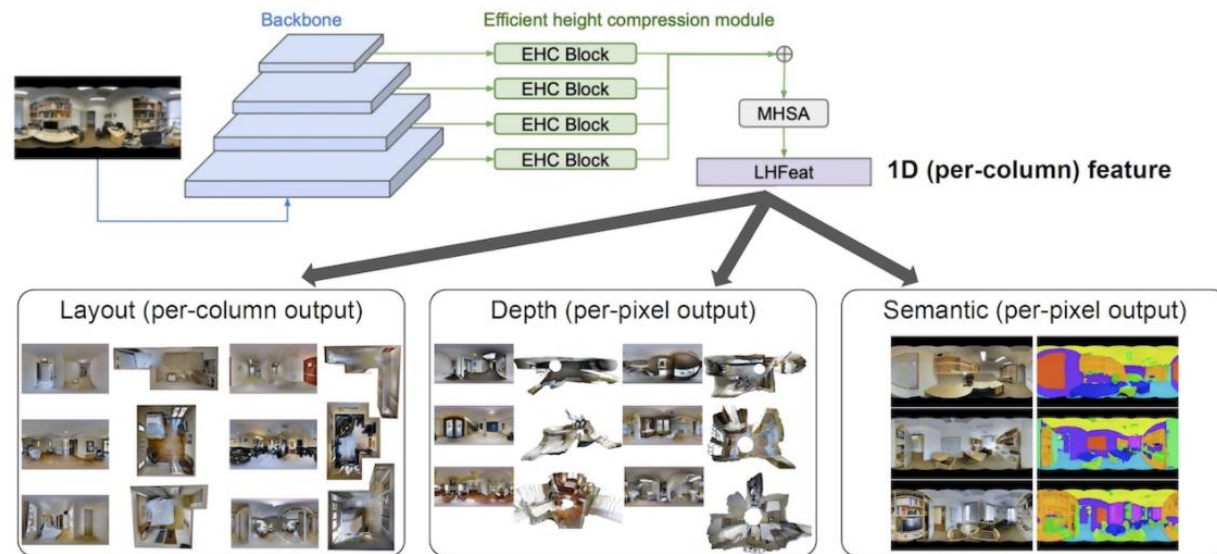
Hohonet은 레이아웃 재구성을 위한 새로운 방법을 설계하는 것이 아님.

Room layout : Hohonet은 resnet34, horizonnet은 resnet50 사용.

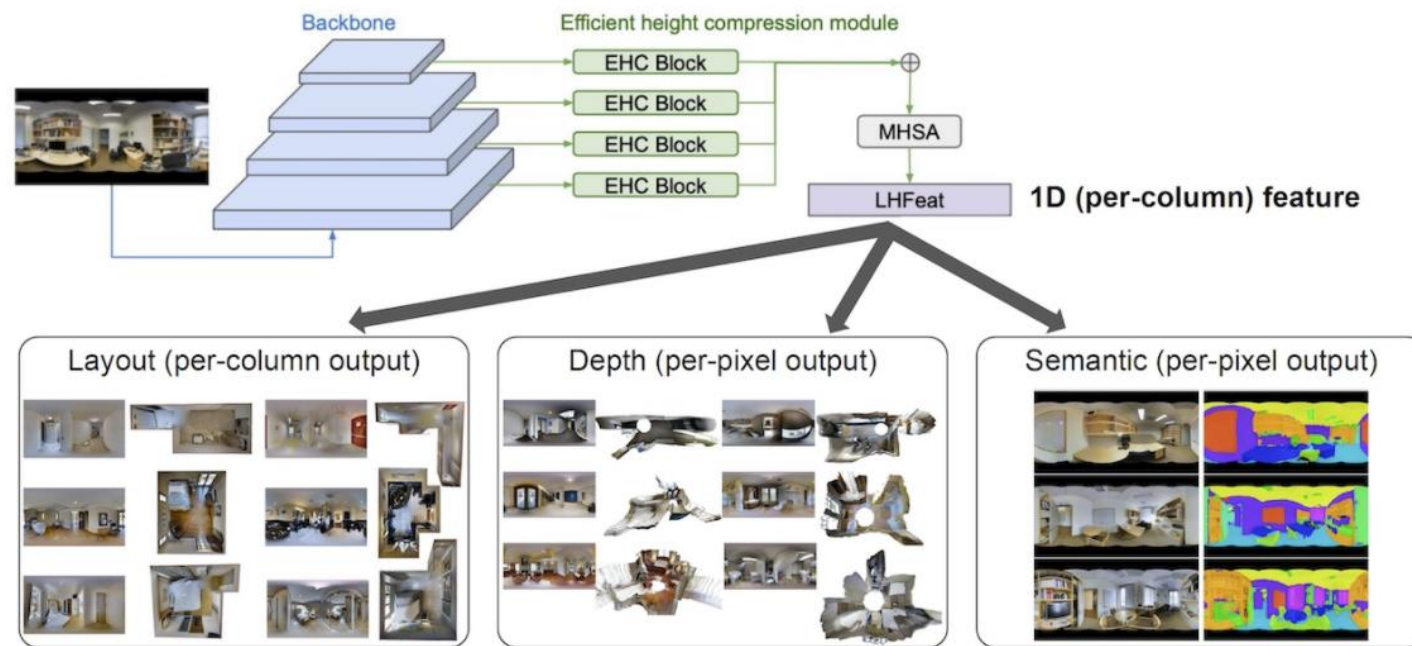
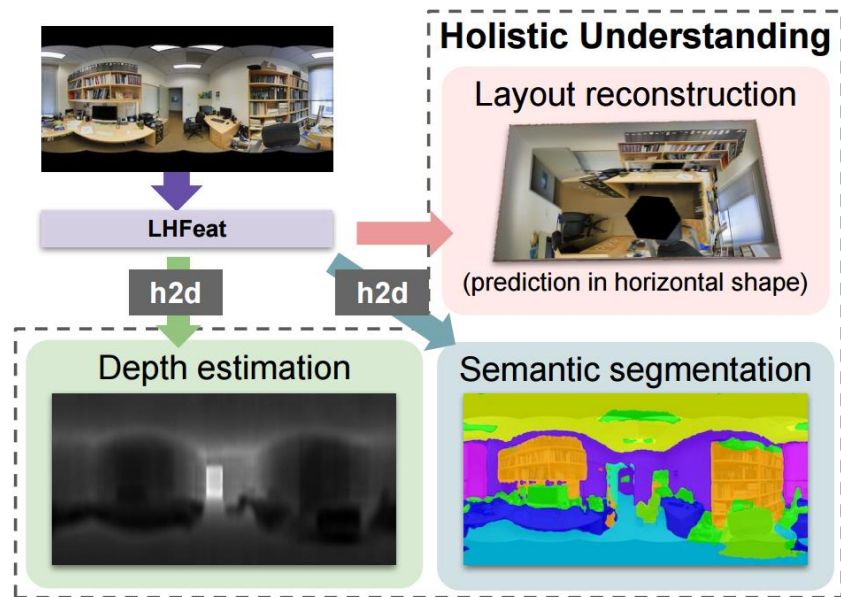
Semantic segmentation : resnet101 사용.

Depth estimation : resnet50 사용

- HoHoNet은 이전의 최첨단 기술인 BiFuse[24]를 큰 폭으로 능가한다는 것을 입증한다. HoHoNet은 장면의 전체적인 구조를 잘 포착한다.
- BiFuse는 ERP와 큐브맵을 모두 모델 입력으로 사용하므로 두 개의 backbone 네트워크가 필요.
- Hohonet 단점 : HoHoNet의 깊이 경계는 BiFuse의 깊이 경계에 비해 더 흐릿하다. 열의 일부 고주파 신호는 HoHoNet에 의해 폐기된다.



4 HoHoNet - 소개



- ① 입력 ERP 이미지는 형상 피라미드 추출을 위해 먼저 CNN backbone을 통과한 다음,
- ② 제안된 효율적인 높이 압축 모듈은 형상 피라미드를 높이 치수가 평평한 잠재 수평 형상 표현(LHFeat)으로 인코딩한다.
- ③ 마지막으로, LHFeat에서 HoHoNet 프레임워크는 최첨단 품질의 열당 및 픽셀당 양식(레이아웃의 모서리 또는 경계)을 모두 제공할 수 있다.



(a) Aligned 360.

(b) Roll rotation.

(c) Pitch rotation.

→ (a)처럼 y축이 중력방향으로 정렬되었을 때 이미지 column 구조 정보를 더 잘 압축하여 보관할 수 있음.

4 HoHoNet - 소개

360 이미지에 대한 깊이 추정.

Depth estimation



전방위 이미지의 깊이를 모델링하기 위해, OmniDepth는 ERP 왜곡을 고려하여 encoder-decoder architecture를 설계한다.

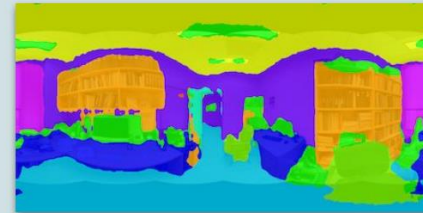
PanoPopups는 평면 인식 손실로 360 깊이를 학습하는 것이 합성 환경에 도움이 된다는 것을 보여준다.

계단식 훈련 단계를 가진 여러 백본(backbone)을 사용하는 대부분의 최신 방법과 대조적으로, **HoHoNet은 하나의 백본으로만 구성되며 한 단계에서만 훈련된다.**

또한, HoHoNet은 **소형 LHFeat을 통해 밀도가 높은 깊이를 모델링**하는 반면 이전의 기술은 기존의 밀도가 높은 특징에서 깊이를 추정한다.

360 이미지에 대한 의미론적 세분화.

Semantic segmentation



의미론적 분할은 장면 모델링의 기본 작업이다.

DistConv는 ERP 이미지의 조밀한 깊이 및 의미 예측을 위한 왜곡 인식 변형 가능한 컨볼루션 레이어를 제안한다. 360 의미 있는 분할을 위한 최근의 대부분의 방법은 정이십면체 mesh와 관련된 표현으로 작동하는 훈련 가능한 층을 설계한다.

그러나 위의 모든 방법은 파노라마 신호에 대해 비교적 낮은 해상도로 실행된다.

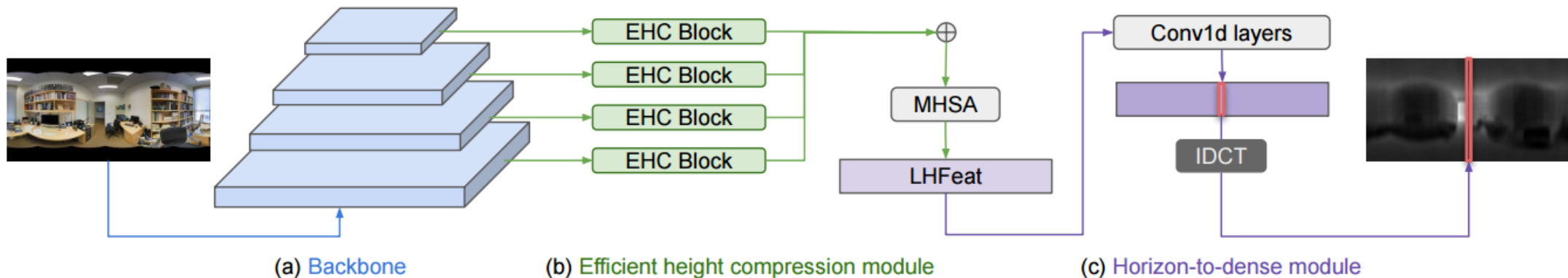
탄젠트 이미지는 고해상도 파노라마를 처리하고 미리 훈련된 가중치를 투시 이미지에 배치할 수 있는 세분화된 20면체에 접하는 다중 평면 이미지에 전방위 신호를 투사한다.

탄젠트 이미지와 마찬가지로 HoHoNet도 **고해상도 이미지에서 작동**할 수 있으며, 이는 더 나은 의미 있는 분할 정확도를 달성하는 데 필수적인 요소로 나타났다.

최근의 방법과 대조적으로 HoHoNet은 **ERP 이미지에서 직접 실행**되며 고도로 최적화된 딥 러닝 라이브러리는 모든 작업을 쉽게 구현할 수 있다.

4

HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ① 전체 구성



① 고해상도 파노라마는 먼저 backbone(예: ResNet)에 의해 처리된다.

② 형상 피라미드는 제안된 EHC(Efficient Height Compression) 모듈과 정교화를 위한 다중 헤드 자기 주의(MHSA) 모듈에 의해 압착 및 융합된다.

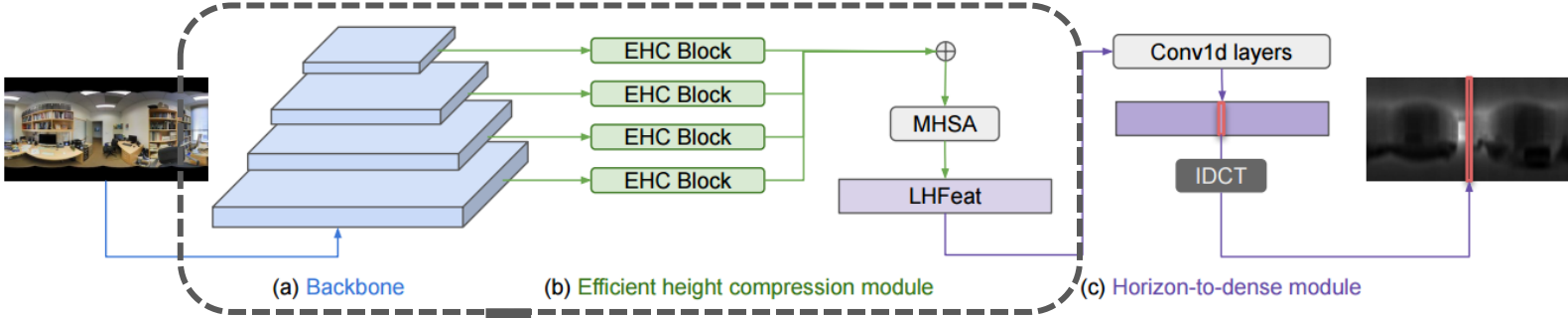
그 결과 LHFeat은 compact며(예: 입력 이미지가 R 3×512×1024인 경우 R 256×1024), 전체 네트워크가 기존의 인코더-디코더 네트워크보다 훨씬 빠르게 조밀한 기능을 실행할 수 있다는 점에 유의한다.

③ 마지막으로, 최종 예측을 산출하기 위해 1D 컨볼루션 레이어를 사용한다.

우리는 DCT 주파수 영역에서 예측이 우수한 결과를 가져온다는 것을 발견하여 각 열의 예측에 IDCT를 적용한다.

4

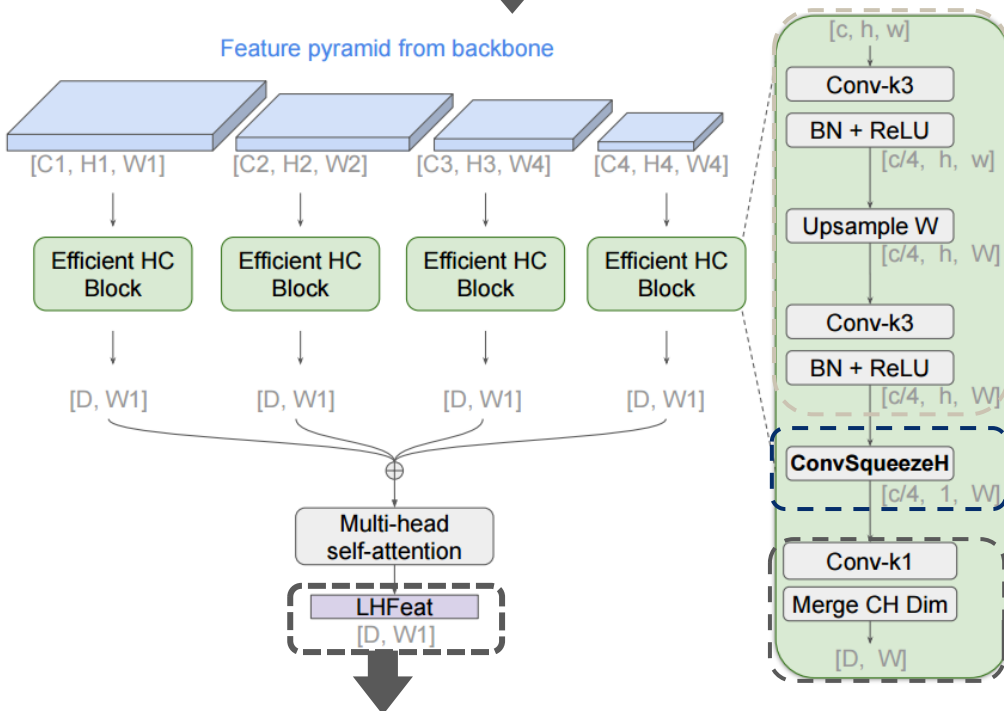
HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ② LHFeat에 대한 EHC(효율 높이 압축) 모듈



먼저 backbone의 피라미드에서 각 2D feature의 높이를 짜내기 위해 EHC 블록을 사용한다.

그 결과 1D 형상이 합산으로 간단히 융합된다.

2D 및 1D 형상의 크기는 각각 $[C, H, W]$ 및 $[C, W]$ 로 표기한다.



1

EHC 블록 내에서 입력 2D feature은 채널 감소를 위해 먼저 Conv2D 블록에 의해 처리된 다음,

필요한 경우 공간 너비가 W_1 로 upsampling되고 마지막으로 다른 Conv2D 블록이 upsampling된 feature을 개선한다.

2

feature 높이를 1로 효율적으로 줄이기 위해 커널 크기를 $(h, 1)$ 로 설정하여 패딩 없이 전체 feature 높이를 커버하는 깊이 있는 컨볼루션 레이어인 ConvSqueezeH 레이어를 설계한다.

ConvSqueezeH 레이어는 커널 크기가 패딩 없이 이전에 알려진 입력 형상 높이로 설정된 깊이 별 컨볼루션 레이어로 출력 형상 높이 1을 생성한다.

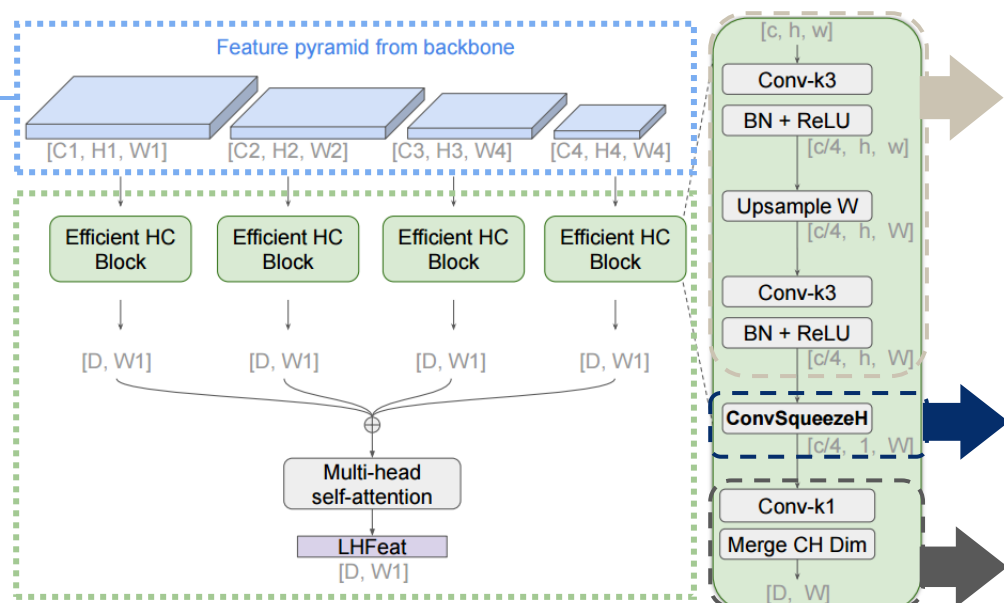
각 EHC 블록의 파라미터 h 는 H_{inp} 가 주어지면 자동으로 사전 계산된다.

3

마지막으로, Conv2D 레이어는 채널 수를 LHFeat의 잠재 크기 D 로 변환하고, 높이 치수는 ConvSqueezeH 레이어에 의해 이미 1로 감소되었기 때문에 간단히 폐기된다.

4

HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ② LHFeat에 대한 EHC(효율 높이 압축) 모듈



1

EHC 블록 내에서 입력 2D feature은 채널 감소를 위해 먼저 Conv2D 블록에 의해 처리된 다음, 필요한 경우 공간 너비가 W_1 로 upsampling되고 마지막으로 다른 Conv2D 블록이 upsampling된 feature을 개선한다.

2

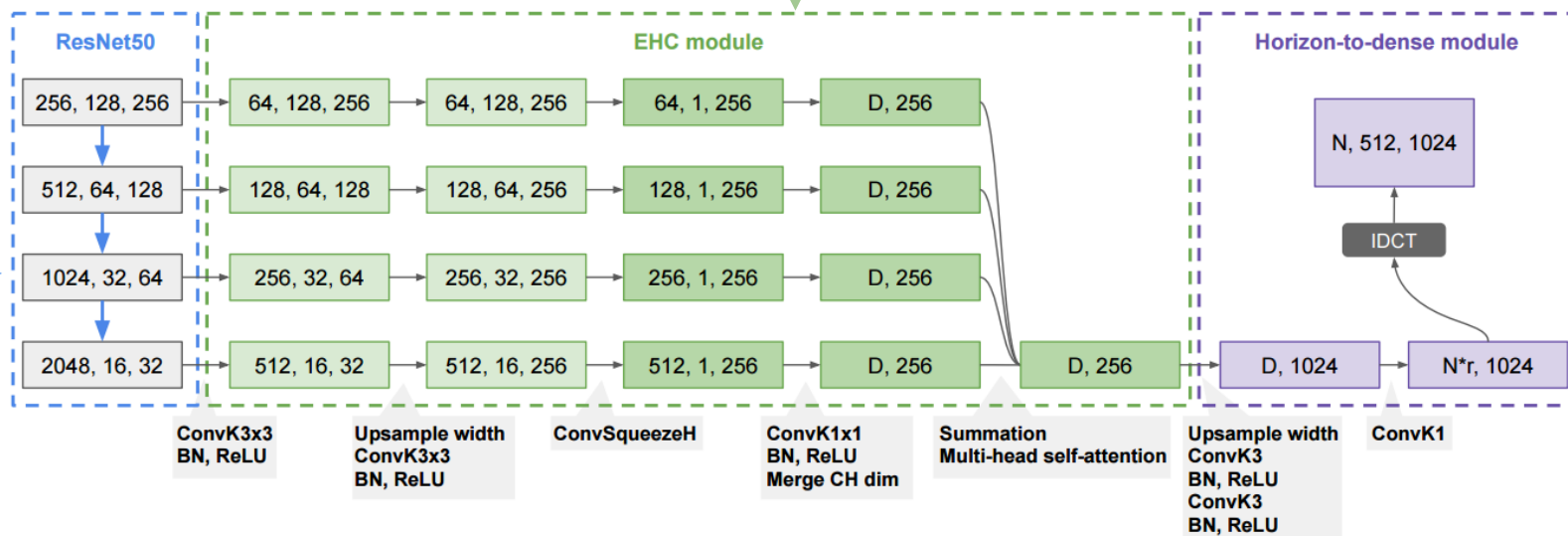
feature 높이를 1로 효율적으로 줄이기 위해 커널 크기를 $(h, 1)$ 로 설정하여 패딩 없이 전체 feature 높이를 커버하는 깊이 있는 컨볼루션 레이어인 ConvSqueezeH 레이어를 설계한다.

ConvSqueezeH 레이어는 커널 크기가 패딩 없이 이전에 알려진 입력 형상 높이로 설정된 깊이 별 컨볼루션 레이어로 출력 형상 높이 1을 생성한다.

각 EHC 블록의 파라미터 h 는 H_{inp} 가 주어지면 자동으로 사전 계산된다.

3

마지막으로, Conv2D 레이어는 채널 수를 LHFeat의 잠재 크기 D 로 변환하고, 높이 치수는 ConvSqueezeH 레이어에 의해 이미 1로 감소되었기 때문에 간단히 폐기된다.



입력 파노라마의 높이와 너비는 각각 512와 1024로 가정한다.

D 와 E 는 하이퍼 파라미터다.

4

HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ② LHFeat에 대한 EHC(효율 높이 압축) 모듈

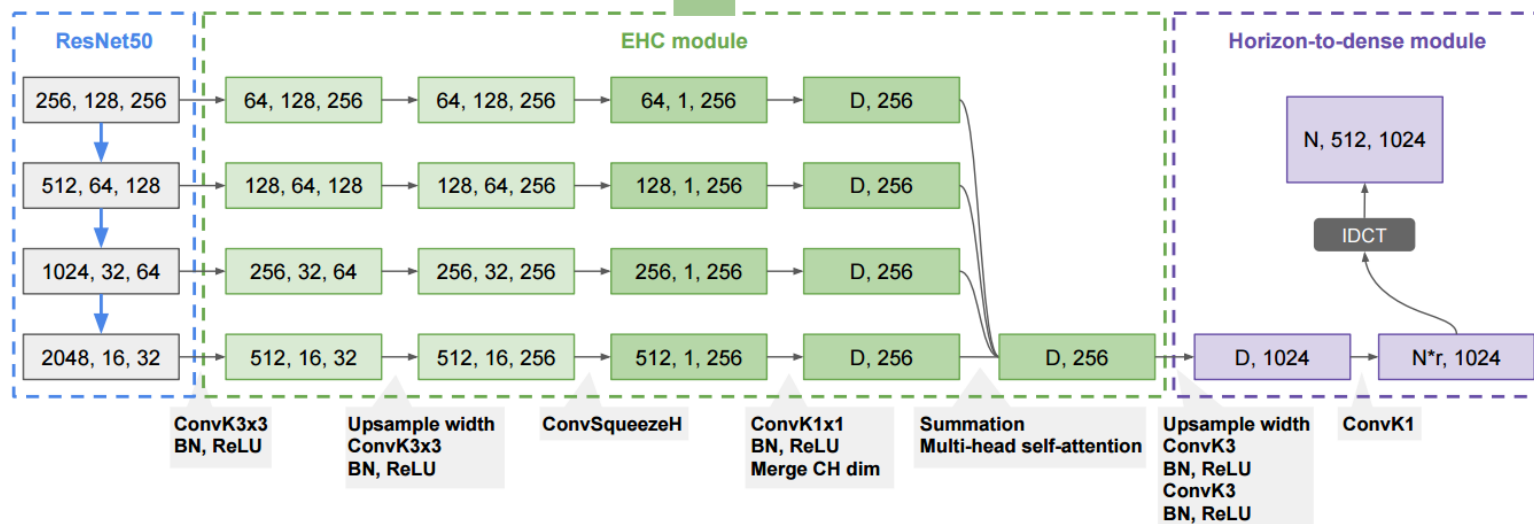
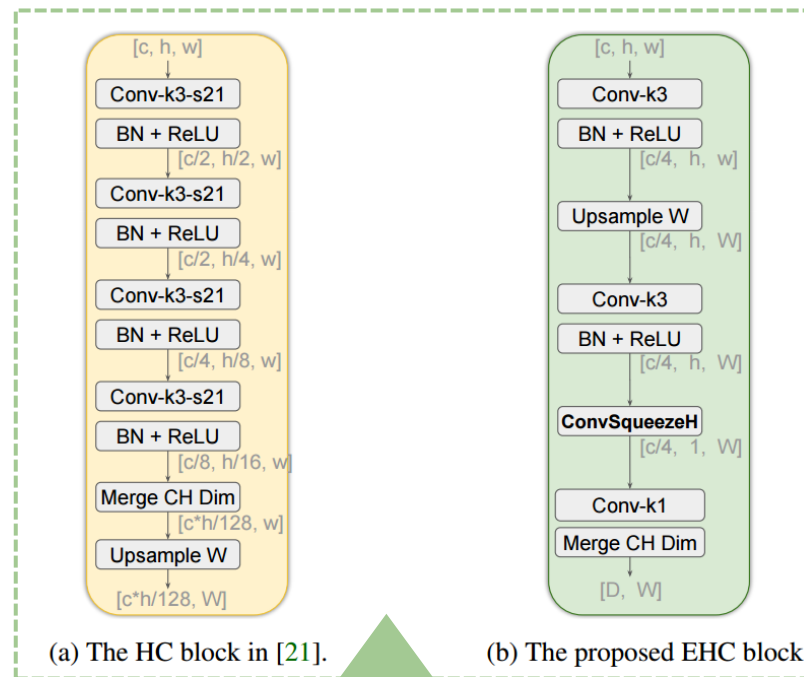
EHC 블록과 HC 블록 비교

높이 압축 블록은 백본에서 2D 형상을 압착하여 1D 수평 형상을 생성하는 것을 목표로 한다.

HC 블록은 일련의 컨볼루션 레이어를 사용하여 채널 수와 높이를 점차적으로 감소시킨다.

반면, EHC 블록은 먼저 채널 감소를 위한 컨볼루션 레이어를 사용한 다음 이중선형 업샘플링 및 ConvSqueezeH 레이어를 사용하여 수평 형상의 형상을 생성한다.

절제 실험에서 HC 블록을 제안된 EHC 블록으로 대체하면 속도와 정확도가 향상된다.



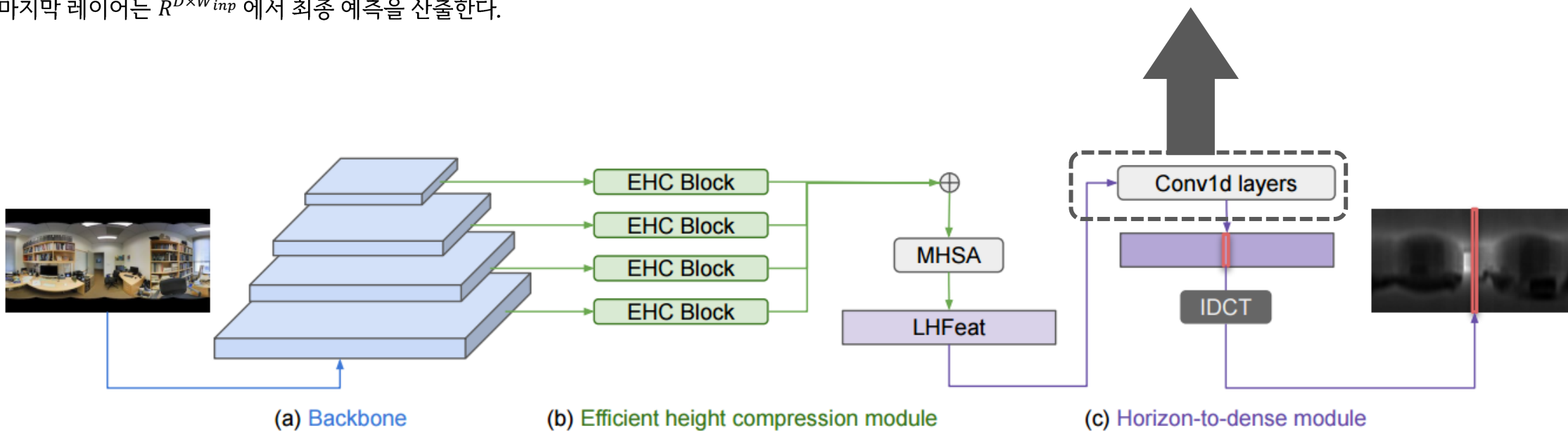
4

HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ③ horizon-to-dense 모듈, 열 당 1D 양식 예측



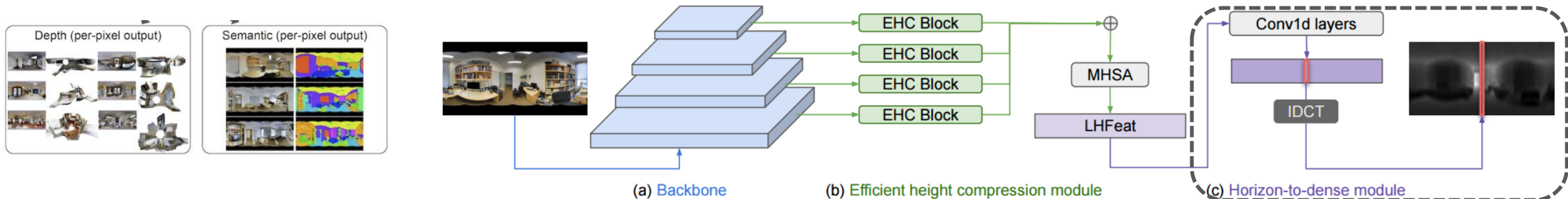
1D 양식을 예측하기 위해 먼저 $R^{D \times W_1}$ 에서 $R^{D \times W_{inp}}$ 로 수평 형상을 upsampling하고 BN, ReLU 사이에 커널 크기 3, 3, 1의 Conv1D 레이어를 각각 적용한다.

마지막 레이어는 $R^{D \times W_{inp}}$ 에서 최종 예측을 산출한다.



4

HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ③ horizon-to-dense 모듈, 픽셀당 2D 양식 예측



출력 공간을 열당 형식으로 shaping하는 전략은 픽셀당 양식이 포함된 작업에는 적용되지 않는다.

여기서는 compact LHFeat $R^{D \times W_1}$ 에서 조밀한 예측 $R^{N \times H_{inp} \times W_{inp}}$ 를 도출하기 위한 HoHoNet의 수평 대 밀도 모듈을 제시한다.

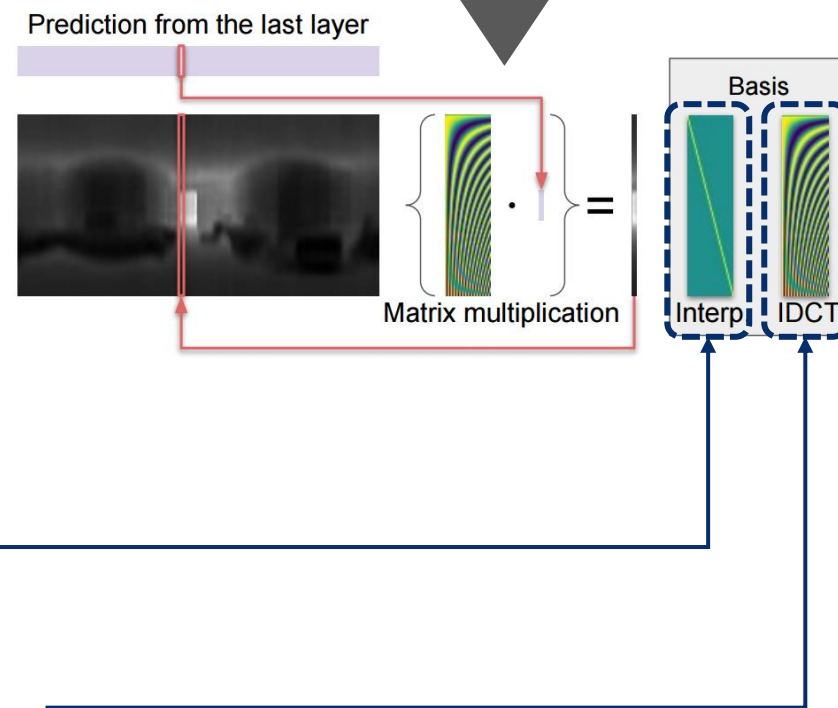
이 기능은 다양한 애플리케이션에 보다 일반적인 시나리오의 문을 열어준다.

2D 양식 예측을 위한 훈련 가능한 계층은 출력 계층의 채널 수가 $E = N \cdot r$ 로 증강되고 여기서 N 은 작업에 대한 대상 채널의 수이고 r 은 이미지 열에 의해 공유되는 구성 요소의 수라는 점을 제외하면 3.3항에서 소개한 1D 예측을 위한 계층과 거의 동일하다.

생성된 예측은 $R^{E \times W_{inp}}$ 에서 $R^{N \times r \times W_{inp}}$ 로 재구성된다. 예측된 r 값에 할당한 물리적 의미에 따라 각 열에 대해 R^r 를 $R^{H_{inp}}$ 로 복구하기 위한 두 가지 다른 연산을 제시한다.

① 보간법 (interpolation)

② 역 이산 코사인 변환 (IDCT, Inverse Discrete Cosine Transform)



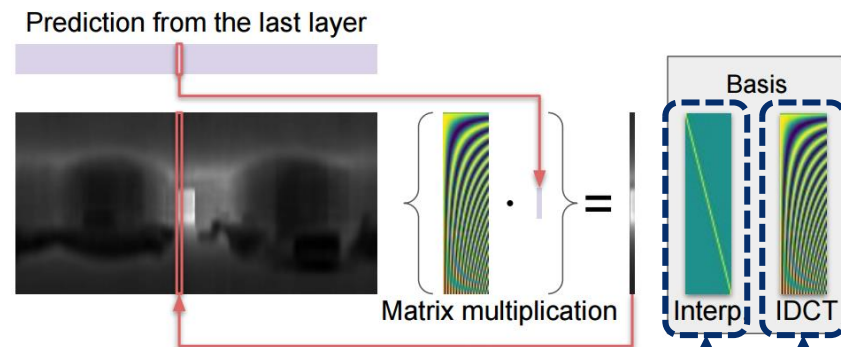
4

HoHoNet - 조밀한 깊이 추정을 위한 HoHoNet 프레임워크 개요 ▶ ③ horizon-to-dense 모듈, 픽셀당 2D 양식 예측

제안된 수평 대 밀도(h2d) 모듈은 compact LHFeat에서 밀도 예측을 생성할 수 있다.

Linear interpolation을 IDCT로 대체함으로써 조밀한 예측 결과를 개선할 수 있다.

수평 대 밀도 모듈(h2d)을 통해 효율적으로 인코딩된 LHFeat은 이제 조밀한 양식을 모델링할 수 있다.



각 열의 예측은 기초 M 의 성분들의 선형 조합에 대한 가중치로 작용한다.

① 보간법 (interpolation)

가장 간단한 방법은 잠재 치수 r 을 출력 높이로 보고 선형 보간법을 적용하여 $r < H_{inp}$ 일 경우, H_{inp} 의 r 크기를 조정하는 것이다.

① HoHoNet은 M 이 선형 보간을 구현하는 경우 공간 영역에서 예측하고

② 역 이산 코사인 변환 (IDCT, Inverse Discrete Cosine Transform)

에너지 압축 특성에 대한 이미지 압축에서 DCT의 적용에 영감을 받아, r 예측 값을 높은 주파수가 잘리는 DCT 주파수 영역에 있는 것처럼 본다.

② M 이 IDCT를 구현하는 경우 주파수 영역에서 학습한다.

이 경우 IDCT를 적용하여 low-pass 신호를 원래 신호로 복구할 수 있다.

IDCT가 선형 보간법을 지속적으로 능가한다.

LHFeat은 공간-행 정보를 혼합하므로, 평평한 행이 없는 LHFeat에서 행에 의존하는 밀도 양식을 분리하기 위해 마지막 층을 훈련시키는 것은 문제가 될 것이다.

반대로, 주파수 영역에서 예측하는 법을 배우는 것은 각 열의 원래 행 정보를 전체적으로 특징짓는 의미 있는 공간 주파수를 가진 잘 정의된 기본 함수로부터 이익을 얻을 수 있으므로 행 의존성 문제를 완화시킬 수 있다.

4

비교 정리



ENTER THE CONTENTS

I believe that someone like you, I was broken my
heart. But now, I am standing again.



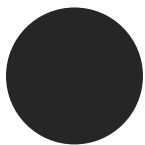
Someone

I believe that someone like you. Enter
something here.



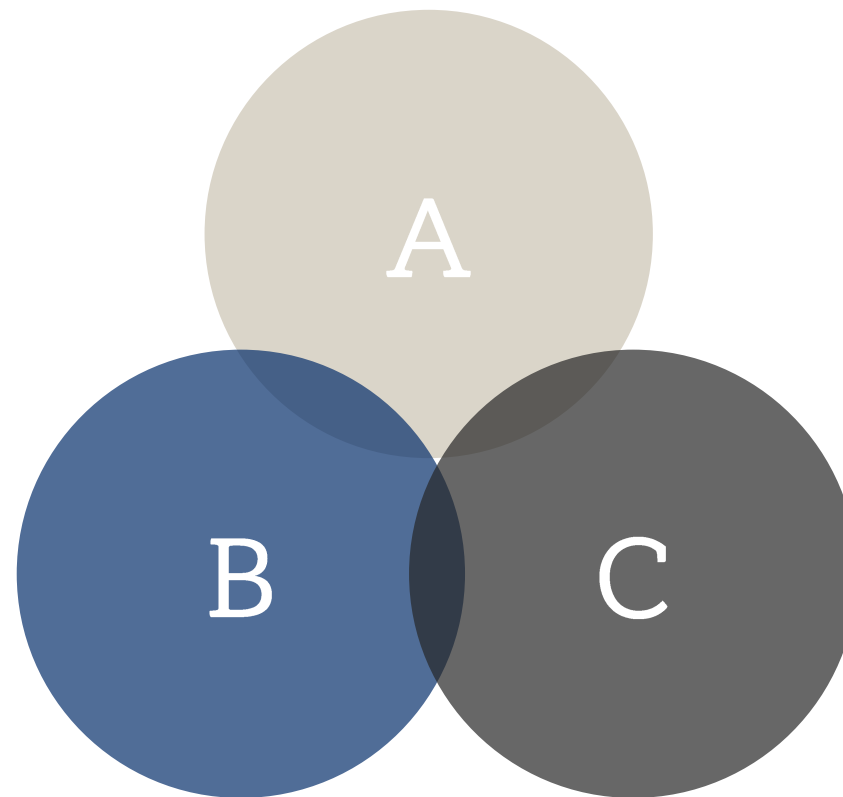
Someone

I believe that someone like you. Enter
something here.



Someone

I believe that someone like you. Enter
something here.



1 CONTENTS

ENTER THE CONTENTS

I believe that someone like you. I was broken my heart. But now, I am standing again.



Something

I believe that someone like you.
Enter something here.

Something

I believe that someone like you.
Enter something here.

Something

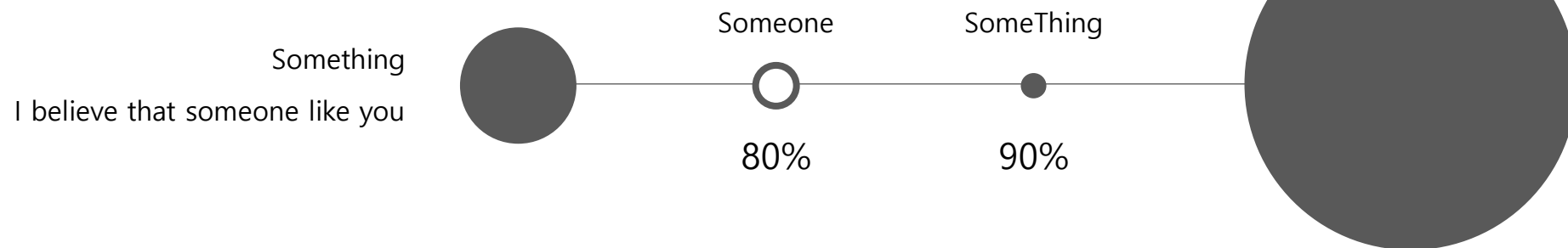
I believe that someone like you.
Enter something here.

2 CONTENTS

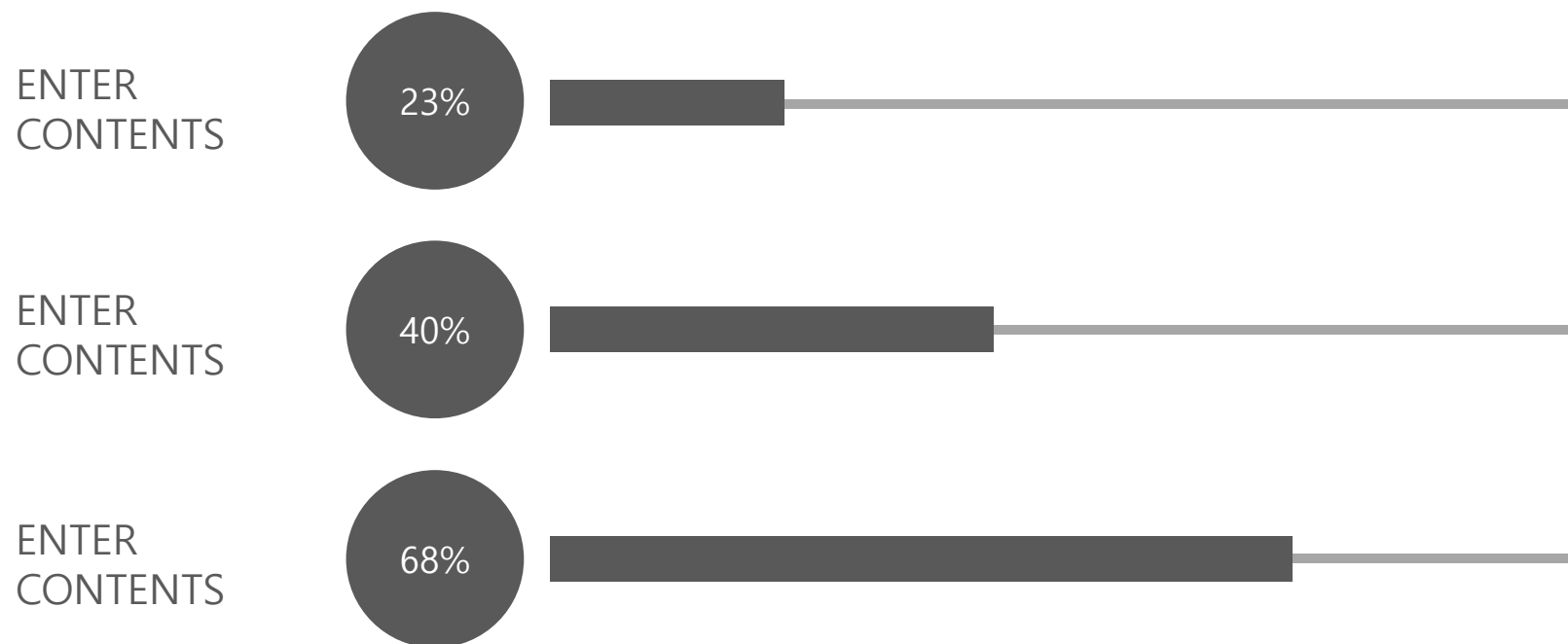
ENTER THE CONTENTS

I believe that someone like you

ENTER



3 CONTENTS



I want you to use this template for free and to remember slug and CREBUGS for me.

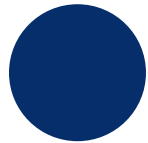
ENTER THE CONTENTS

I believe that someone like you. I was broken my heart. But now, I am standing again.



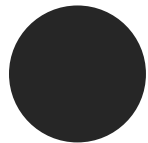
Someone

I believe that someone like you.
Enter something here.



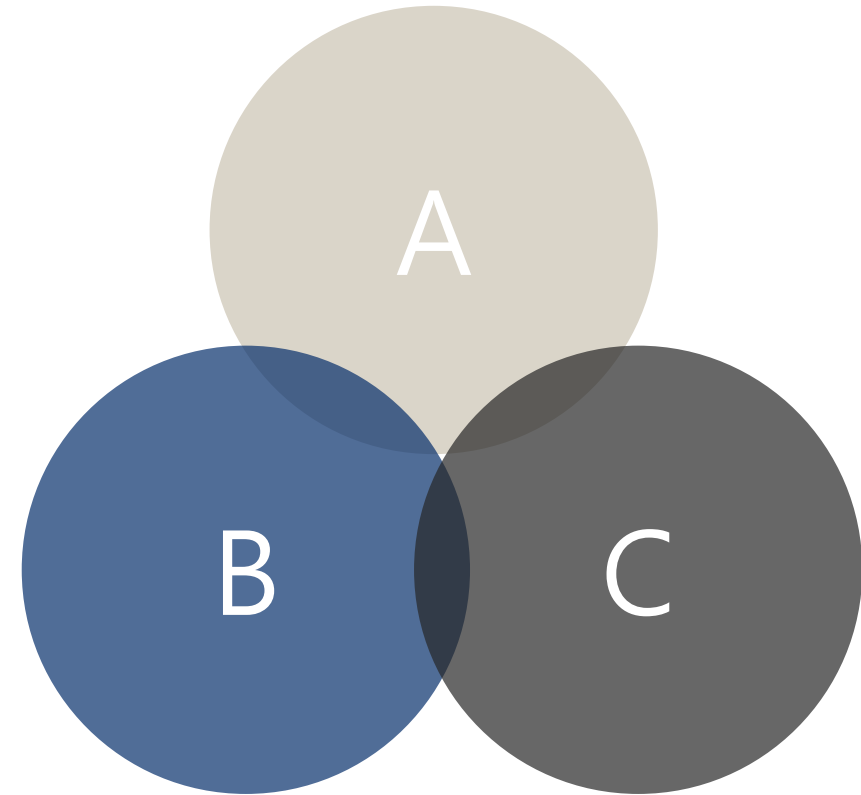
Someone

I believe that someone like you.
Enter something here.



Someone

I believe that someone like you.
Enter something here.





ENTER THE CONTENTS

I believe that someone like you. I like pizza.
I already know that you want to meet me.

CONTENTS 29%



ENTER THE CONTENTS

I believe that someone like you. I like pizza.
I already know that you want to meet me.

CONTENTS 70%



ENTER THE CONTENTS

I believe that someone like you. I like pizza.
I already know that you want to meet me.

CONTENTS 50%



ENTER THE CONTENTS

I believe that someone like you. I like pizza.
I already know that you want to meet me.

CONTENTS 98%





THANK YOU SO MUCH!