공학석사 학위논문

효율적인 야외환경 의미론적 영상분할을 위한 깊이정보 활용 방법

Exploiting Depth information for efficient Outdoor Semantic Segmentation

2023년 2월

서울대학교 대학원

전기 정보 공학부

우 명 우

효율적인 야외환경 의미론적 영상분할을 위한 깊이정보 활용 방법

Exploiting Depth information for efficient Outdoor Semantic Segmentation

지도 교수 서 승 우

이 논문을 공학석사 학위논문으로 제출함

2023년 2월

서울대학교 대학원

전기 정보 공학부

우 명 우

우명우의 공학석사 학위 논문을 인준함

2023년 2월

위 원 장:	조 남 익	(인)
부위원장:	서 승 우	(인)
위 원:	김 성 우	(인)

국문초록

Semantic segmentation은 이미지를 이해하는 가장 포괄적인 방법이다. 이미지의 모든 픽셀에 분류하고자 하는 semantic 클래스 레이블을 부여하는 것이다. 자율 주행 및 로봇 등을 운용하는 관점에서 인식의 영역을 담당하기 때문에 매우 중요한 기술이다. 최근에는 RGB 이미지뿐만 아니라 깊이 정보를 추가로 이용해서 Semantic segmentation의 성능을 향상하려는 시도가 이루어지고 있다. 하지만 대부분의 시도는 야외보다는 실내 환경에서 많이 시도되고 있다. 또한 쉽게 깊이 정보를 활용하기 힘든 부분들이 있다. 그 이유는 첫째, 야외에서는 정확하고 밀도 높은 깊이 정보를 얻는 것이 상대적으로 더 어렵다. 둘째, 깊이 정보를 network의 입력으로 처리할 경우 추가적인 인코더를 요구하기 때문에 새로운 구조의 네트워크 설계가 필요하다.

본 논문에서는 앞서 언급한 어려움을 극복하고, 효율적인 깊이 정보 사용을 위한 방안을 모색한다. 이를 위해서 "깊이 및 픽셀 위치 기반 어텐션(**D**epth and **P**ixel-distance based **A**ttention : **DPA**) 모듈"을 제안한다. 이 모듈은 깊이 정보를 활용해서 픽셀 사이의 상관관계를 추론하는 데 사용한다. 픽셀의 클래스 유사성은 동일한 객체에 속하는 픽셀이 유사한 깊이 값을 갖는다는 사실을 이용하여 계산된다. 깊이의 상대적 차이만 고려되기 때문에 제공된 깊이 정보의 정확성에 대해서 상대적으로 강건하다. 또한, **DPA**는 기존의 RGB 기반 segmentation 네트워크에 적용할 수 있는 간단한 플러그인 모듈이다. 깊이 정보처리를 위해서 새로운 네트워크 설계가 필요로 하지 않고 기존에 잘 작동하는 RGB 기반의 네트워크에 쉽게 적용이 가능하다. 또한 깊이 정보 처리를 위한 추가적인 인코더가 필요하지 않기 때문에 계산 측면에서도 훨씬 효율적이다. **DPA**는 깊이정보를 입력정보로 활용하지 않고, RGB 기반의 feature에 깊이 정보를 간접적으로 제공한다. 이를 통해서 기존의 RGB 기반의 feature를 강화한다.

다양한 baseline 네트워크에 **DPA**모듈을 적용해서 성능과 효율성을 검증하였다. baseline 모델의 종류와 관계없이, Semantic segmentation의 성능을 개선하였고, 깊이 정보를 입력으로 활용하는 기존 방식에 비해서 연산량 측면에서 효율적임을 검증하였다.

주요어: 의미론적 영상분할, 언텐션, 인공신경망, 딥러닝
학 번: 2021-21575

목차

표 목차

그림 목차

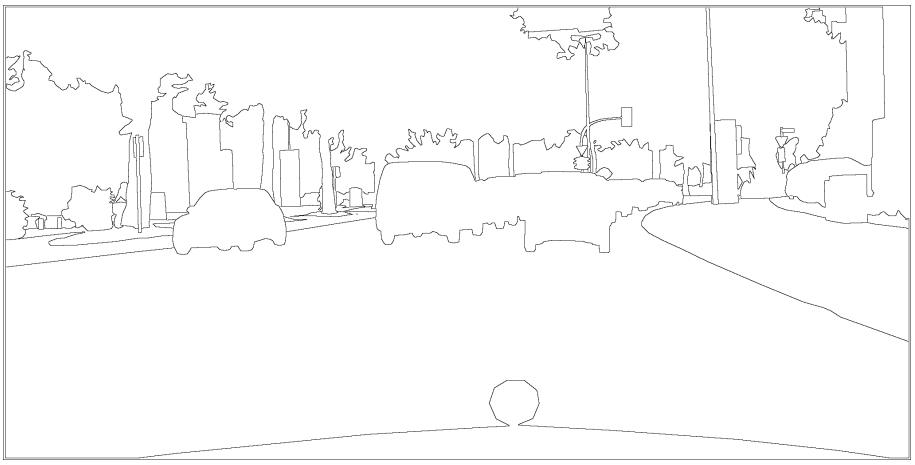
제 1 장 서 론

Semantic segmentation은 이미지의 모든 픽셀에 클래스 레이블을 할당하는 작업이다. Deep Convolutional Neural Network(DCNN)의 개발과 함께 상당한 발전이 있었으며, [1], [29]은 FCN[12] 기반 접근 방식으로 많은 성공을 거두었다. Semantic segmentation은 컴퓨터 비전의 기본적이고 필수적인 분야이며 자율 주행, 로봇 공학, 의료 이미지 처리와 같은 다양한 분야에서 사용된다. 자율 주행이나 로봇이 작동하는 상황에서는 깊이 정보가 3D 재구성, 위치 파악, 주행거리 측정 등 다양한 목적으로 제공되거나 생성되는 경우가 많다. 깊이 정보는 이미지에서 생략된 추가 기하학적 정보이다. 이미지는 3D 실제 환경에서 2D로 투영된 정보이며, 그 과정에서 정보 손실이 발생한다. 깊이 정보는 이미지에서 프로젝션 과정에서 발생한 정보 손실을 보상할 수 있다. 그러나 이러한 장점에도 불구하고 다음과 같은 이유로 실내 환경에서 주로 연구되어 왔다.

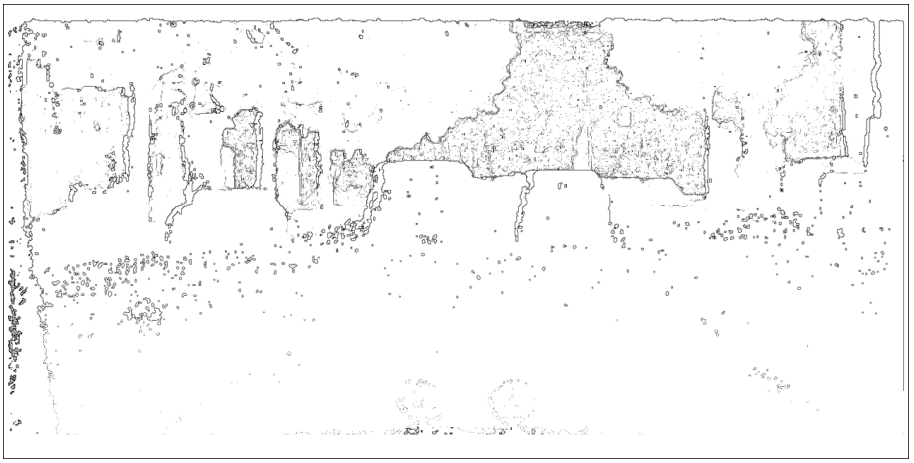
1) RGBD 센서의 야외에서의 성능 제한. RGBD 센서는 검출 거리가 10m 미만으로 상대적으로 짧고, IR 센서를 사용하기 때문에 야외에서 햇볕을 쬐면 간섭이 발생한다. 따라서 대부분의 RGBD 데이터 세트는 실내 환경으로 제한된다. 최근에는 스테레오 카메라와 딥 러닝 기반의 모노 카메라가 야외에서 저비용으로 깊이 정보를 획득하기 위해 사용되고 있다. 이렇게 얻은 깊이 정보는 RGBD 센서보다 밀도가 낮고 정확도가 떨어지지만 잘 활용하면 충분히 Semantic segmenatation에 도움이 될 수 있다. 2) 새로운 입력신호를 사용하려면 새로운 네트워크 구조가 필요하다. FuseNet[6], RedNet[11], ACNet[8] 및 ESANet[15]와 같은 네트워크는 깊이 정보를 입력으로 받아들인다. 이러한 네트워크는 깊이 정보 입력을 위해 병렬 인코더를 사용한다. 이 전략은 RGB 및 깊이 인코더에서 나온 feature를 융합하여 더 나은 좋은 feature를 생성하는 것이다. 이 방법은 직관적이지만 기존 RGB 기반 네트워크를 사용할 수 없고

새로운 네트워크 구조의 설계가 필요하다. 새로운 네트워크 구조를 설계하는 것은 어려운 작업이다. 또한, 인코더가 중복적으로 사용되기 때문에 계산량이 현저히 증가한다.

본 논문에서는 이전에 잘 사용되지 않았던 야외환경에서 깊이 정보를 활용하는 데 중점을 둔다. 이를 위해, 우리는 "깊이 및 픽셀 위치 기반 어텐션 (**D**epth and **P**ixel-distance based **A**ttention : **DPA**) 모듈"을 제안한다. 깊이 정보를 입력으로 사용하는 것이 아니라 픽셀 간의 상관관계를 찾기 위한 정보로 사용한다. 이는 동일한 레이블에 속하는 픽셀은 유사한 깊이 값을 가진다는 가정에 기초한다. 사람은 깊이 정보를 통해서 물체의 대략적인 윤곽을 볼 수 있기 때문에 깊이 정보를 보고 물체의 레이블을 추론할 수 있다. 깊이 정보의 윤곽선은 깊이 값의 불연속점을 나타낸다. 즉, 깊이 값이 급격히 변하는 부분이다. 물체의 경계선이 아닌 부분은 깊이 값이 부드럽게 변하고, 부드럽게 변경되는 부분은 동일한 객체 내의 부분이다. **그림 1.1**은 레이블 및 깊이 정보에서 값 변화가 큰 부분을 Laplacian 필터를 통해 추출한 결과를 보여준다. 깊이 정보에 noise가 많지만, 경계선은 비슷 경향을 보인다. 이는 동일한 레이블에 속하는 픽셀이 다른 레이블에 속하는 픽셀과 비교하여 유사한 깊이 값을 갖는다는 것을 보여준다. 이러한 특성을 이용하여, 깊이 값의 유사성은 픽셀들 간의 유사성으로 정의될 수 있다. 그러나 야외환경에서 시야가 넓을 경우 유사한 깊이가 항상 동일한 레이블을 갖는 것은 아니다. 유사한 깊이 값을 가진 다른 레이블의 물체도 존재할 수 있다. 만약 깊이 유사성이 **그림 1.2 (b)**와 같이 깊이에 의해서만 계산된다면 동일한 깊이에 있는 다른 여러 물체에 대해 유사성이 높게 나온다. 추론하고자 하는 픽셀(query)에서 멀리 떨어져 있지만 깊이 값이 비슷한 물체가 유사성이 높게 나와서 원하는 곳에 집중하지 못한다. 이 문제를 해결하려면 제약 조건이 필요하다. 깊이 값이 비슷하더라도 너무 멀리 떨어져 있는 점은 동일한 객체에 속하지 않는다. 사람은 깊이 값뿐만 아니라 값이 속한 위치까지 관찰하기 때문에 깊이 맵을 보고 사물을 인식할 수 있다. 즉, 깊이 기반 유사성은 특정 거리 내에 있을 때 충족된다. 따

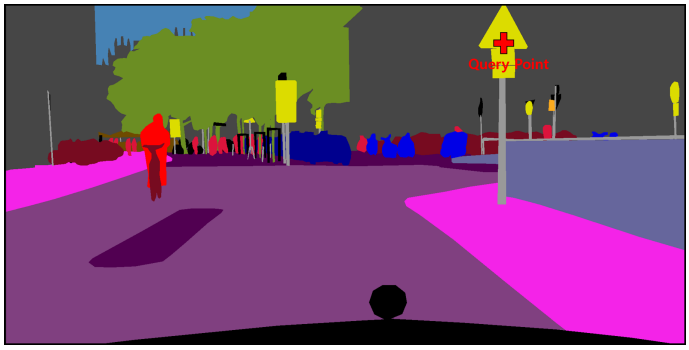


(a)레이블에 Laplacian 필터 적용 결과.

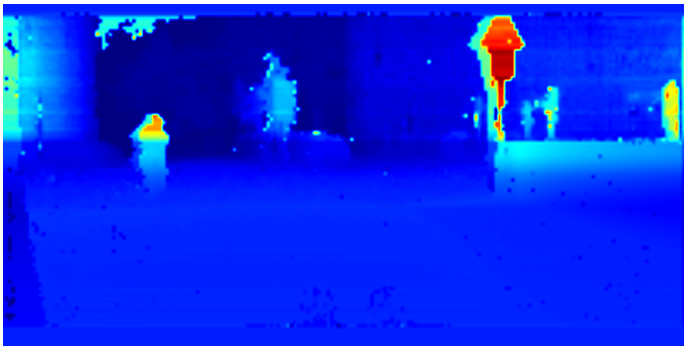


(b)깊이 정보에 Laplacian 필터를 적용한 결과.

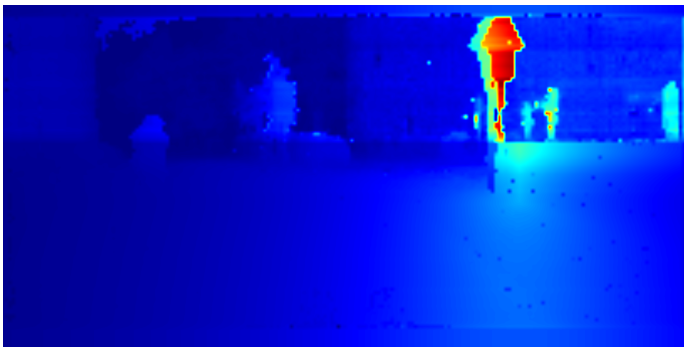
그림 1.1: 경계선 추출결과



(a)Ground Truth와 query 지점



(b)깊이 정보만 활용한 유사도 시각화



(c)깊이 정보 및 픽셀 위치 정보를 활용한 유사도 시각화

그림 1.2: 깊이 유사도 시각화

라서 깊이 유사성을 계산할 때 픽셀 사이의 거리가 고려된다. 유사도 계산에 픽셀 위치를 포함하면 **그림 1.2 (c)**와 같이 추론하고자 하는 픽셀(query)과 같은 물체에 해당하는 영역이 더 집중된다. 이 연산은 깊이 정보의 각 점에 수직 및 수평 픽셀 위치를 추가하여 3D 유클리드 공간을 생성한다. 이 3D 유클리드 공간에서의 거리는 픽셀 간의 유사성으로 정의된다.

이 정의를 바탕으로, **DPA**는 backbone 네트워크를 통해 추출된 feature에 대해서 attention 및 feature aggregation을 수행한다. **DPA** 모듈에 깊이 정보가 입력되면 픽셀 위치가 추가되어 3D 유클리드 공간이 형성되며, 이 3D 공간에서의 인접성을 기준으로 픽셀 간의 유사성이 결정된다. 이러한 유사성을 바탕으로 query인 픽셀과 유사성이 높은 픽셀에 높은 attention 가중치가 할당된다. 결과적으로, attention 가중치는 깊이 값이 비슷하고 거리가 가까울수록 더 높은 가중치를 갖는다. 우리는 이것을 "깊이 및 픽셀 위치 기반 어텐션 가중치"라고 부른다. feature aggregation은 깊이 및 픽셀 위치 기반 어텐션 가중치를 통하여 수행한다. 이러한 방식으로 feature에 깊이 정보를 간접적으로 포함시킨다. 이 방법은 가공되지 않은 깊이 정보를 통해 attention 가중치를 계산하기 때문에 깊이 정보를 활용하기 위해 새로운 인코더를 필요로 하지 않는다. **DPA** 모듈을 통해서 새로운 네트워크를 설계하지 않고, 기존 RGB 기반 네트워크에 추가하는 방식으로 깊이 정보의 활용이 가능하다. 또한, 추가적인 인코더가 필요하지 않기 때문에, 깊이를 입력으로 사용하는 것보다 연산량의 증가가 크지 않아서 효율적이다.

제 2 장 배 경 지 식

2.1 Semantic segmentation

Semantic segmentation은 컴퓨터비전의 가장 기본적이고 핵심적인 분야 중 하나이다. 이미지를 분류하는 것이 아닌, 이미지를 픽셀 단위로 이해해야 하는 높은 수준의 문제이다. 우리가 정의한 의미론적 단위로 분류하는 작업으로, 픽셀 단위의 Classification으로 볼 수 있다. **그림 2.1**은 Semantic segmentation의 입력과 출력을 보여준다. 입력으로 RGB 또는 흑백의 정보가 들어가게 되면, 출력으로 픽셀이 어느 클래스 레이블인지 나타내 준다. 조금 더 자세히 살펴보면, 네트워크의 출력은 One-Hot encoding으로 각 클래스에 대한 채널의 출력값이 softmax의 확률로 출력이 된다. 하나의 픽셀에서 가장 높은 확률값을 가지는 채널에 해당하는 클래스로 추정된다.

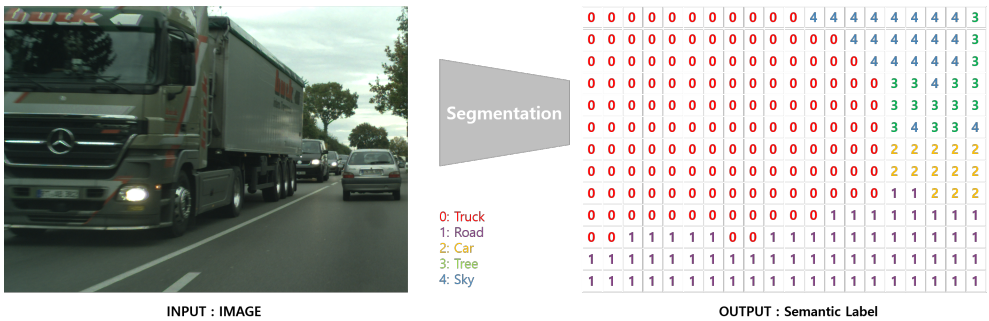


그림 2.1: Semantic segmentation 예시

Semantic segmentation도 다른 컴퓨터 비전 분야처럼 Convolutional Neural Network(CNN)의 발달과 함께 많은 발전을 이루어냈다. 하지만, Image classification과는 다르게 Semantic segmentation은 픽셀들의 위치정보의 보존이 필수적이다. CNN의 특성상 layer를 거듭하면서 이미지의 해상도는 줄이고, 채널의 수는 늘리면서 feature를 추출한다. 이 과정에서 위치정보가 많이 손실되기 때문에, 추출된 feature를 바로 사용하기가 힘들다. 그래서 다시 입력 이미지와 동일한 해상도로 Up-sampling 하는 과정이 필요하다. **그림 2.2**은 일반적인 segmentation 네트워크의 구조이다. 앞서 언급한 대로 해상도를 낮추(Down-sampling)면서 feature를 추출하는 부분인 인코더 부분이 있고, 추출된 feature를 바탕으로 해상도를 높이(Up-sampling)면서 클래스를 추정하는 디코더 부분이 있다. FCN[12]이 가장 대표적인 CNN을 활용한 방식의 Semantic segmentation이다. 기존의 Fully Connected(FC) layer를 사용하던 것을, 1×1 convolution layer로 바꾼 네트워크이다. 오늘날의 대부분의 Semantic segmentation 네트워크들은 기본적으로 FCN[12]과 유사한 구조를 유지한다. 다양한 형태의 Semantic segmentation 네트워크가 개발되고 있지만, 인코더/디코더의 구조는 유지되고 있다. 의료영상 분야에서 제안되어서 다양한 분야에서 활용 중인 U-Net[14]의 경우도, 앞선 인코더와 디코더 구조를 유지하고 있다. 최근에는 더욱 좋은 feature 추출을 위해서 더욱 강력한 인코더를 적용하거나, Transformer[17]를 인코더에 이용하는 네트워크[21]도 있다. 또한 Semantic segmentation에 특화된 인코더를 제안하는 HRNet[18], STDC[4] 등 다양한 시도가 이루어지고 있다.

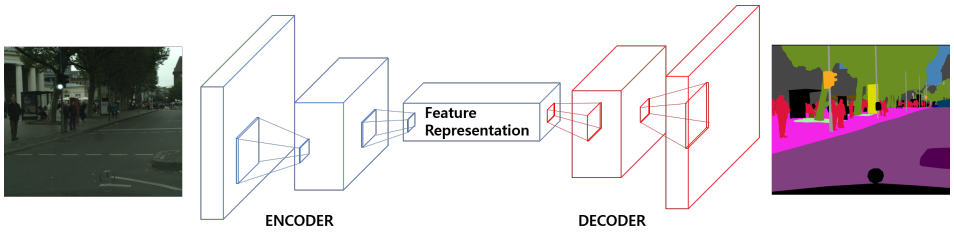


그림 2.2: 인코더/ 디코더 구조의 Semantic segmentation 네트워크

2.2 RGBD Semantic segmentation

Semantic segmentation 네트워크는 크게 두 부분으로 나눌 수 있다. 입력 정보로부터 feature를 추출하는 인코더 부분과 추출된 feature를 기반으로 각 픽셀의 클래스를 추론하는 디코더 부분이다. 깊이 정보를 Semantic segmentation에 활용하기 위한 다양한 시도가 있지만 대부분은 깊이 정보를 추가 입력으로 사용한다. 깊이 정보를 추가 입력으로 활용하기 위해서는 새로운 인코더를 필요로 한다. **그림 2.3**과 같이 RGB 정보를 위한 인코더와 깊이 정보를 위한 인코더를 병렬로 사용하고, 각각의 인코더에서 추출된 feature를 혼합함을 통해서 더 나은 추론 결과를 만들어낸다. FuseNet[6]은 이름에서 알 수 있듯이 깊이 정보 feature와 RGB feature를 융합한다. 병렬 인코더를 통해 FuseNet[6]은 깊이 정보와 RGB를 추출 및 융합하여 더 나은 feature를 생성한다. FuseNet[6]와 유사하게 RedNet[11]도 깊이 정보 인코더에서 추출된 feature를 RGB feature와 융합한다. ACNet[8]은 융합을 위한 모듈과 인코더를 별도로 사용하는 것을 제외하고는 이전 방법과 동일한 기본 전략을 사용한다. ESANet[15]도 병렬 인코더를 사용하고 모바일 로봇을 위한 실시간 네트워크를 제안한다.

또 다른 방법은 깊이 정보를 사용하여 CNN 작동 방식을 변경하는 것이다. Depth-aware CNN[19]의 경우, CNN 및 pooling 연산을 수행할 때, 깊이 값이 유사한 픽셀을 사용에 더욱 높은 가중치를 부여하여 조정된다. 또한 Malleable 2.5D convolution[22] 또는 3D neighborhood convolution[2]은 위의 아이디어

를 기반으로 CNN 연산을 2D 이상으로 확장한다. 앞서 언급한 방법들은 대부분 실내 환경에 초점을 맞추고 있다. 또한 깊이 정보를 활용하기 위해서는 새로운 네트워크의 설계가 필요하다. 이것은 기존의 RGB 기반 네트워크를 사용할 수 없다는 것을 의미한다. 본 논문에서 제안하는 **DPA**는 깊이 정보를 사용하여 픽셀 간의 상관관계를 추론하고 이를 통해서 feature를 생성한다. 그렇기 때문에 이 방법은 깊이 정보에서 feature를 추출하기 위해 추가적인 인코더나 네트워크가 필요하지 않다. 따라서, **DPA** 모듈은 깊이 정보를 입력으로 사용하지 않기 때문에 큰 구조적 변화 없이 성능이 우수한 기존 RGB 기반 네트워크를 이용해서 Semantic segmentation 성능을 향상시킬 수 있다.

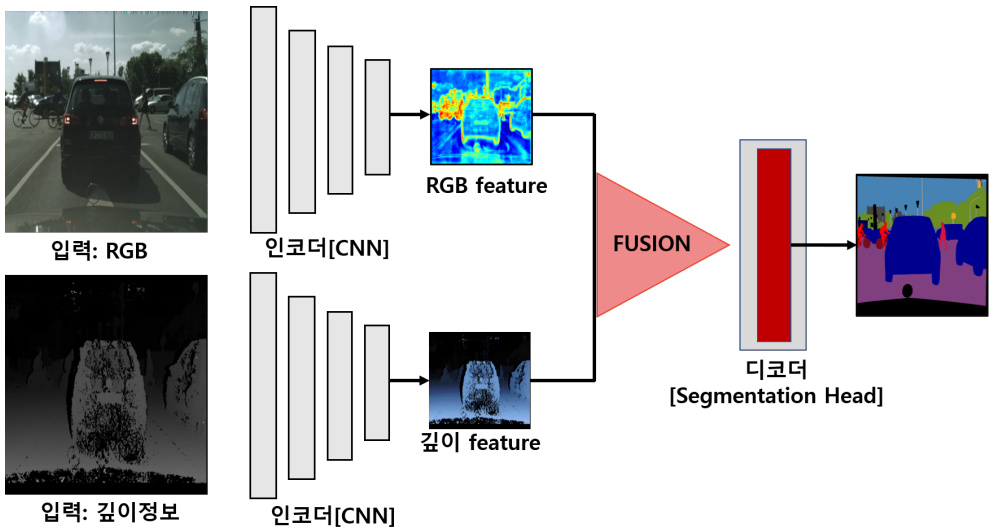


그림 2.3: RGBD Semantic segmentation 네트워크(병렬 인코더)

2.3 Context aggregation

Semantic segmentation은 각 픽셀에 레이블을 부여하는 임무이지만, 하나의 픽셀을 결정하려면 주변 픽셀도 고려해야 한다. 따라서, 주변 픽셀의 context 정보를 aggregation 하는 것은 정확한 Semantic segmentation을 수행하기 위해서 중요한 작업이다 . FCN[12]는 Semantic segmentation에서 상당한 성공을 거두었지만, 작은 receptive field에서 충분한 context 정보를 추출하는 데 어려움이 있다. 이러한 문제를 해결하기 위하여 다양한 크기 또는 scale의 receptive field를 적용하려는 시도가 있었다. PSPNet[29]은 spatial pyramid pooling을 사용하여 서로 다른 pooling layer에서 다양한 scale의 context 정보를 추출한다. DeepLabv2[1]은 Atrous Spatial Pyramid Pooling(ASPP)를 통한 연산량 증가를 최소화하면서 receptive field를 확장하고 다양한 receptive field의 feature를 융합한다. **그림 2.4**은 Atrous Convolution의 작동원리를 보여준다. Dilation rate는 얼마나 넓은 영역을 처리할지를 결정하는 것으로, 일반적인 convolution은 dilation rate = 1인 것으로 볼 수 있다. Dilation rate가 넓어질수록, receptive field는 넓어지지만, 연산에 참여하는 픽셀의 수는 동일해서 아무리 영역이 넓어져도 3x3 convolution과 연산량은 동일하다. ASPP는 다양한 Dilation rate를 통해서 추출된 feature들을 이용해서 다양한 scale의 정보를 추출한다. DenseASPP[23]의 경우, ASPP에 밀도 높은 연결, 즉 DenseNet[9]의 아이디어를 추가하여 다양한 scale의 정보를 융합한다.

최근에는 Attention을 기반으로 context 정보를 추출하려는 여러 시도가 있었다. Non-local neural network[20]은 모든 픽셀 간의 상관관계를 계산하여 attention 가중치를 생성한다. **그림 2.5**은 Non-local neural network[20]의 attention 가중치 계산을 시각화한 그림이다. 인코더로 부터 추출된 feature 기반으로 query와 key를 생성하고, 그림과 같이 query 픽셀과 key의 모든 픽셀과의 관계를 계산한다. 이를 통해서 만들어지는 attention 가중치는 feature 차원을 $\mathbb{R}^{c_i \times h \times w}$이라고 하면, 모든 픽셀 사이의 관계를 나타내야 하므로 $\mathbb{R}^{h \times w \times h \times w}$

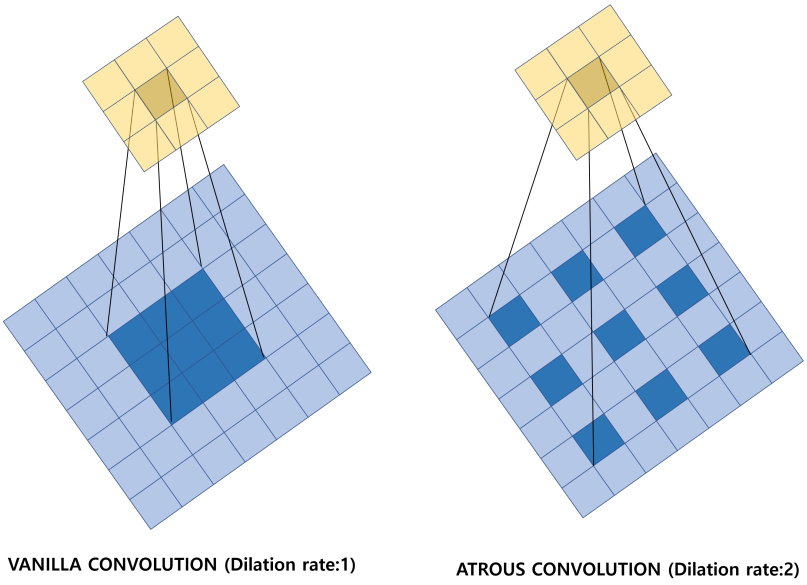


그림 2.4: Atrous Convolution의 구조

가 된다. 이렇게 만들어진 attention 가중치는 픽셀 사이의 context 정보를 담고 있다. 이 가중치를 기존 feature와 행렬 곱을 통해서 context정보를 aggregation 한다.

DANet[5]은 각 픽셀 간의 상관관계뿐만 아니라 feature의 채널 간의 상관관계도 계산한다. CCNet[10]의 경우 Non-local neural network의 계산 비용을 줄이기 위해 criss-cross(교차 경로) attention을 수행한다. **그림 2.6**는 Non-local[20] 방식과 criss-cross[10]방식의 차이를 시각화한 것이다. Non-local[20]의 경우 하나의 query 픽셀에 대해서 key의 모든 픽셀사이의 관계를 계산하지만, criss-cross[10]는 십자영역에 해당하는 픽셀만 상관관계를 계산한다. 이를 통해서 Non-local[20]에 비해서 연산량을 크게 감소 시켰다. [27], [28], [26]도 feature에 대한 픽셀 간의 상관관계를 다양한 방식으로 추론하여 더 풍부한 context 정보를 가진 feature로 강화했다. 본 논문에서도 픽셀 간의 상관관계를 추론하고 context 정보의 aggregation을 수행한다. 이전 방법은 인코더로부터 추출된

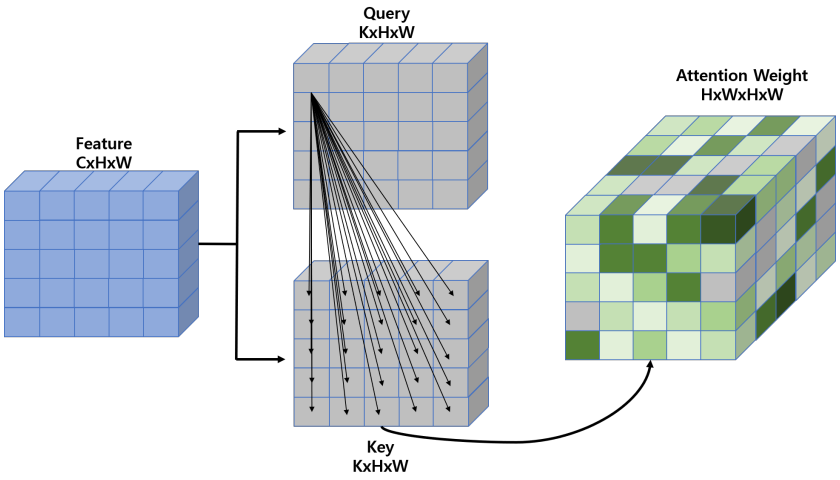


그림 2.5: Non-local neural network의 attention 가중치 계산

feature를 query로 사용하여 상관관계를 추론하지만, 우리의 방법은 깊이 정보를 query로 사용하여 픽셀의 상관관계를 추론한다. 이 과정을 통해 RGB 정보만 있는 feature는 깊이의 context 정보가 포함된 feature로 변환된다.

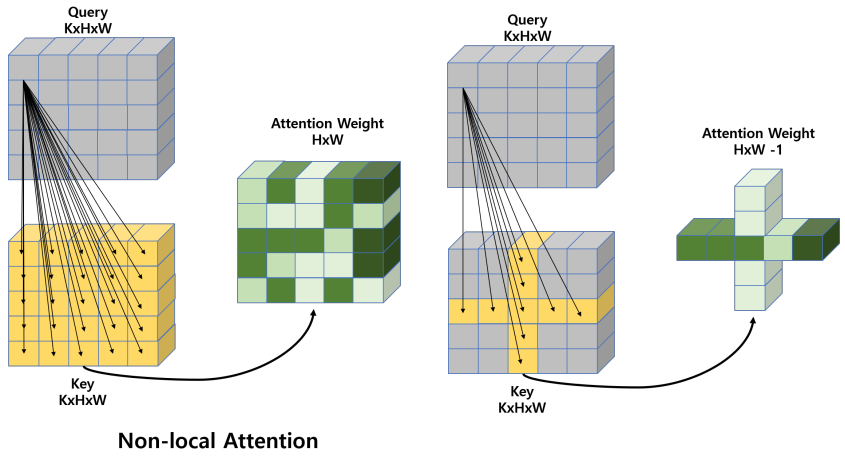


그림 2.6: Non-local Attention과 Criss-Cross Attention의 차이

제 3 장 제 안 방 법

3.1 Baseline 네트워크

본 논문에서는 “깊이 및 픽셀 위치 기반 어텐션(Depth and Pixel-distance based Attention: **DPA**)모듈”을 기존의 RGB 기반의 네트워크에 적용해서 깊이 정보를 활용한다. baseline으로 사용된 네트워크는 BiSeNetV2[24], STDC[4], HRNet[18]이다. 이 네트워크들은 RGB 정보를 처리하기 위한 Semantic segmentation 네트워크로, 인코더/ 디코더의 구조로 이루어져 있다. 이 절에서는 각각의 네트워크의 기본구조와 특성을 설명한다.

3.1.1 BiSeNetV2

BiSeNetV2[24]은 실시간 Semantic segmentation 네트워크이다. 이름에서 알 수 있듯이 BiSeNet[25]의 개선 모델이다. 두 버전 간의 기본적인 설계개념은 동일하지만, V2의 경우 이전 버전에 비해서 네트워크의 비효율적인 부분을 찾아서 제거하고, 단순화 시켰다. BiSeNetV2[24]의 기본적인 아이디어는 다음과 같다. 기존의 실시간 Semantic segmentation 네트워크들은 속도를 향상시키기 위해서 low-level detail을 포기하는 방식으로 설계되었다. 하지만 Semantic segmentation은 모든 픽셀에 대한 추론이 필요한 임무로, low-level detail과 high-level semantics가 모두 중요하다. 기존 네트워크에서 손실되던 low-level detail을 보존하기 위해서, 네트워크의 구조를 2개의 branch로 설계한다. **그림 3.1**과 같이 풍부한 low-level detail의 정보를 추출하기 위해서 Detail branch에서는 보다 깊은 채널의 feature를 추출하되 layer의 깊이는 줄인다. Semantic branch에서는 채널은 상대적으로 작지만, 깊은 layer를 추가해서 Receptive field를 넓히고, 더욱 풍부한 Semantic 정보가 담긴 feature를 추출한다. 각각의 branch에서 추출된 feature 들의 fusion을 통해서 상호보완적인 feature

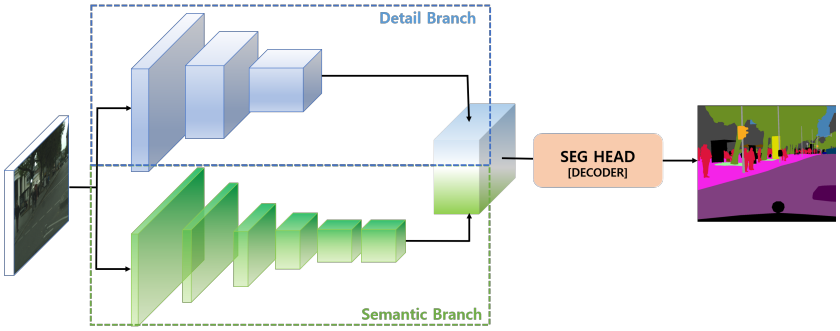


그림 3.1: BiSeNetV2의 네트워크 구조

를 만들어낸다. 이 feature를 디코더(Segmentation Head)를 통과시켜서 최종 추론 결과를 추출한다. 이러한 네트워크 구조를 통해서 연산속도를 높이면서도, 만족할 만한 성능을 만들어낸다.

3.1.2 STDC

STDC[4]의 논문 제목은 “Rethinking BiSeNet For Real-time Semantic segmentation”으로 BiSeNet[25]의 구조를 심층적으로 분석하고, 효율성과 성능을 개선한 논문이다. STDC[4]에서는 low-level detail의 정보를 보존하기 위해서 2개의 Branch를 사용하는 것은 비효율적이라고 주장한다. 이 비효율을 해결하기 위해서, BiSeNet[25]의 Detail Branch를 제거하고, 이를 보완할 수 있는 인코더를 제안한다. Short-Term Dense Concatenate(STDC)모듈은 각 layer의 다양한 scale의 정보를 효율적으로 취합한다. STDC 네트워크는 STDC 모듈로 이루어져 있고, 앞서 언급했듯이 STDC 모듈이 다양한 scale의 정보를 효율적으로 취합하기 때문에, Detail branch와 같은 추가적인 branch가 없이도

높은 성능을 보여준다. 또한 부족한 low-level의 정보는 초기 layer의 feature를 최종 feature와 fusion을 통해서 보완한다. **그림 3.2**은 BiSeNet[25]과 STDC[4]의 네트워크 구조의 차이를 나타낸 그림이다. STDC[4]는 Detail branch(spatial path)를 제거하였기 때문에 연산량을 크게 감소시켰다. 그뿐만 아니라 효율적인 STDC 모듈 구조로 인해서 성능도 더 뛰어나다.

STDC[4]는 STDC1과 STDC2의 두 가지 버전의 인코더를 제공한다. 이것은 STDC 모듈을 얼마나 깊게 쌓아서 만든지에 따른 분류이고, STDC2가 더욱 깊은 네트워크이다. STDC2가 네트워크가 깊은 만큼 연산량은 증가하지만, 성능은 더욱 뛰어나다.

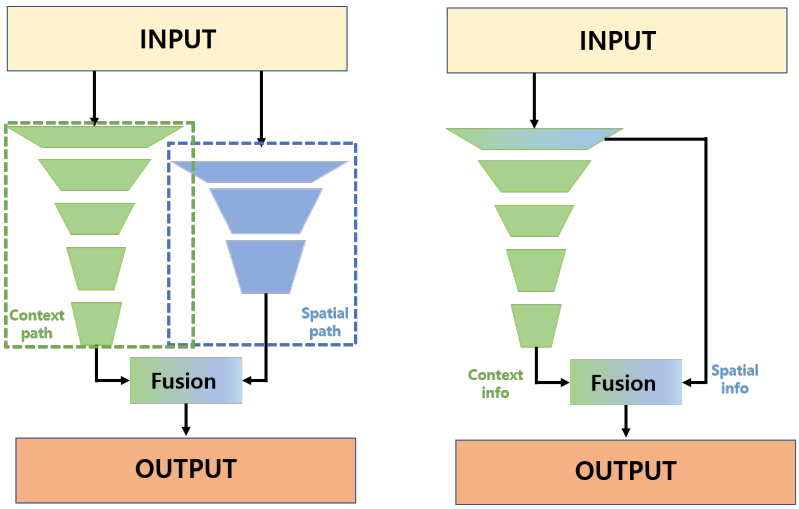


그림 3.2: BiSeNet과 STDC 네트워크 구조차이

3.1.3 HRNet

HRNet[18]은 추론 속도보다 성능에 중점을 둔 네트워크이다. Semantic segmentation의 특성상 다른 컴퓨터 비전 임무에 비해서 고해상도의 정보 구현이 더욱 중요하다. FCN[12]이후 대부분이 인코더/ 디코더 형식의 네트워크 구조로 되어 있다. 인코더/ 디코더 형식의 특성상 저해상도로 정보를 압축하면서 특징을 추출하는 과정에서 많은 datail 한 정보가 손실된다. HRNet[18]는 인코더의 특징추출 과정에서 고해상도의 정보를 보존하는 방식을 제안한다. 이를 위해서 다른 네트워크의 인코더와 유사하게 layer가 진행되면서 고해상도에서 저해상도로 feature가 축소되는 것을 허용하되, 병렬적으로 네트워크를 구성해서 고해상도의 feature가 유지될 수 있는 구조를 제안한다. **그림 3.3**과 같이 네트워크가 병렬적으로 진행되면서, 고해상도의 정보가 유지되고, 각 해상도의 정보를 상호 간의 교환을 통해서 전반적인 정보와 detail 한 정보를 풍부하게 담은 feature를 추출한다.

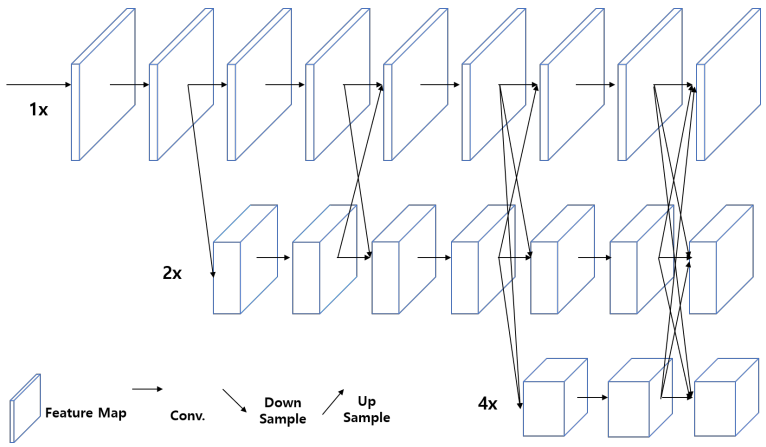


그림 3.3: HRNet의 네트워크 구조

3.2 네트워크 구조

그림 3.5는 본 논문에서 제안하는 **DPA** 모듈을 최신의 Semantic segmentation 네트워크인 BiSeNetV2[24], STDC[4] 및 HRNet[18]에 적용하는 방법을 설명한다. Semantic segmentation 네트워크는 크게 두 부분으로 나누어진다. feature embedding을 수행하는 인코더와 추출된 feature를 기반으로 예측을 수행하는 디코더(Segmentation Head)이다. **DPA** 모듈은 RGB feature를 깊이의 context 정보가 포함된 feature로 강화시키는 과정이다. 따라서 디코더에 들어가기 전에 인코더로 부터 추출된 feature에 대해 "깊이 및 픽셀 위치 기반 어텐션"이 수행된다. **그림 3.4**과 같이, 우리는 1×1 Convolution layer를 통해 RGB feature를 **V**(Value)에 embedding되고, 깊이 정보는 픽셀 위치 정보와 연결되어 **Q**(깊이 Query)를 생성하며, 이는 픽셀 간의 유사성을 계산하는 Query와 Key로 사용된다. 계산된 유사성은 Softmax를 통해 정규화되고 Attention 가중치로 변환되어 **V**에 적용된다. 마지막으로, 초기 RGB feature를 추가하는 잔여 연결이 여기에 적용된다. 깊이의 Context 정보가 포함된 feature를 디코더로 전달한다. 디코더의 경우 3×3 Convolution layer, Batch norm 및 ReLU가 적용된다. 손실 함수는 최종 출력에 Cross-entropy를 적용한다. BiSeNetV2[24], STDC[4]는 baseline 네트워크와 동일하게 Cross-entropy 대신 OHEM[16]을 적용한다.

3.3 깊이 및 픽셀 위치 기반 어텐션(DPA) 모듈

"깊이 및 픽셀 위치 기반 어텐션(Depth and Pixel-distance based Attention: **DPA**)모듈"은 **그림 3.4**에 나와 있듯이, 깊이 정보와 픽셀 사이의 거리를 기반으로 픽셀 간의 유사성을 추론하고, 이를 통해 feature를 강화한다. 이 과정에는 두 가지 입력이 필요하다. RGB 기반의 Segmentation 인코더로부터 추출된 feature $\mathbf{x} \in \mathbb{R}^{c_i \times h \times w}$ 와 깊이 정보 $\mathbf{D} \in \mathbb{R}^{1 \times h \times w}$ 이다. c_i는 추출된 feature의

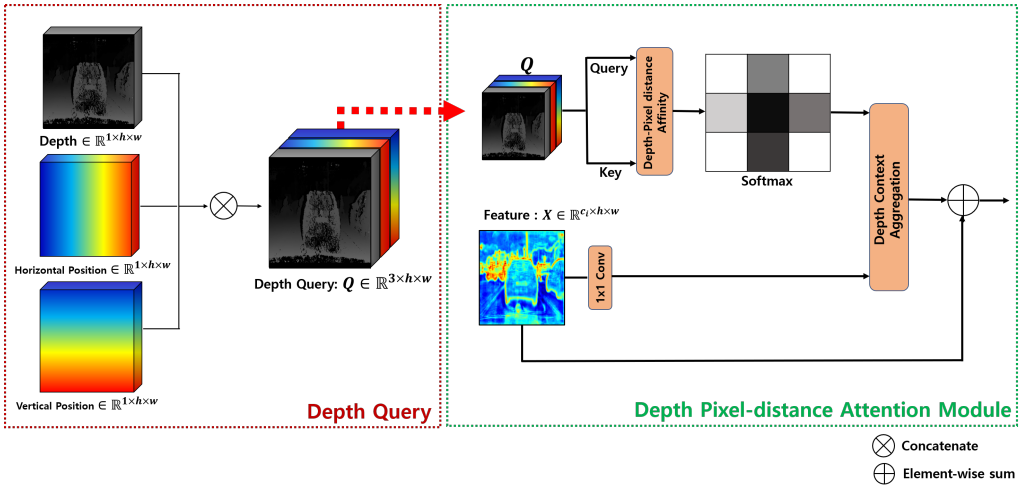


그림 3.4: 거리 및 픽셀위치 기반 어텐션 모듈

채널이고, h 와 w는 인코더를 통과하면서 축소된 feature의 가로와 세로 크기이다. 유사도 계산을 위해서 깊이 정보는 feature의 가로와 세로 크기에 맞춰서 축소한다. 여기서, 픽셀의 위치정보 $\mathbf{P} \in \mathbb{R}^{1\times h\times w}$는 픽셀 사이의 위치를 식별할 수 있도록 깊이 정보와 함께 concatenation 한다. 픽셀의 위치 정보는 [-1,1]의 값으로 정규화한다. 이렇게 만들어진 것을 "Depth query $\mathbf{Q} \in \mathbb{R}^{3\times h\times w}$"로 정의한다. "Depth query($\mathbf{Q}$)"는 깊이 정보, 픽셀의 가로 위치 정보, 픽셀의 세로 위치 정보로 총 3개의 채널로 이루어져 있다. $\mathbf{Q}$는 2D의 픽셀 위치와 깊이 정보를 포함하는 3D 유클리드 공간이다. 유사성은 생성된 $\mathbf{Q}$로부터의 거리를 통해 계산되기 때문에 "Depth key($\mathbf{K}$)"는 $\mathbf{Q}$와 같다. 이렇게 형성된 3D 공간은 3차원의 정보가 포함되어 있지만, 각 차원의 scale이 유사도 계산을 하기에 적합하게 정렬되어 있지 않다. 즉, 각각의 차원이 전체적인 유사도에 얼마나 기여할지에 대한 가중치가 고려되지 않았다. 이를 보정해 주기 위해서 "Scale parameter"를 도입한다. Scale parameter는 각 차원의 축의 가중치를 조절해주기 위해서 적용된다. 이 Scale parameter는 고정된 값이 아니라, 학습의 과정을

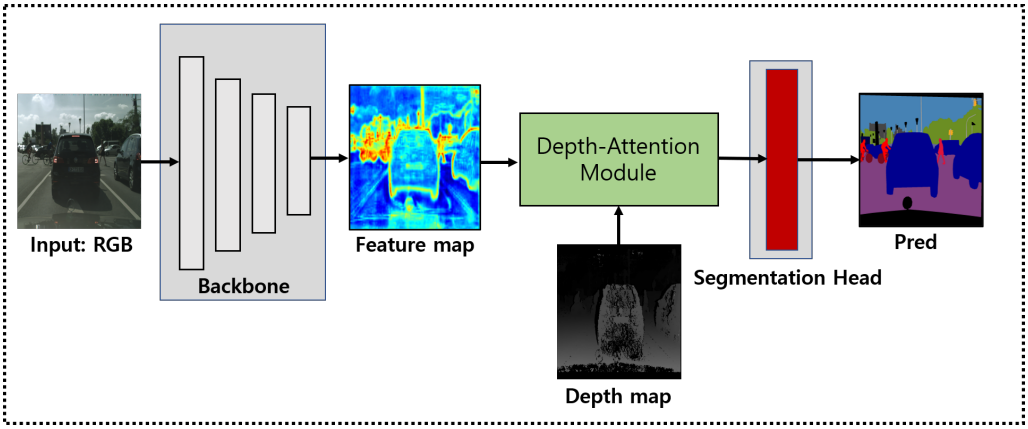


그림 3.5: 거리 및 픽셀위치 기반 어텐션 segmentation 네트워크의 전체적인 구조

통해서 각각의 가중치가 최적화될 수 있는 Learnable parameter이다. Depth query(**Q**)의 임의의 위치를 p_i라고 할때, 이에 대응하는 3차원 벡터는 **(3.1)**과 같이 정의된다.

$$Q(p_i) = (\alpha_d d, \beta_x x, \gamma_y y) \tag{3.1}$$

위 식에서 d는 깊이 정보, x는 가로 위치, y는 세로 위치이다. α, β, γ는 각각의 차원의 축에 해당하는 Scale parameter이다. 이 Scale parameter는 학습 과정에서 end-to-end로 최적화된다. 이러한 과정을 통해서 만들어지는 **Q**는 3D 유클리드 공간이다. 픽셀 간의 유사도에 대한 연산은 **Q**를 기반으로 이루어진다. 유사도는 **Q**에 의해서 생성된 3D 유클리드 공간 내에서의 인접성(Adjacency)이다. 각각의 픽셀에 대해서 유사도를 계산하고, 이것을 attention 가중치로 만들어주기 위해서 Softmax를 통해서 정규화해 준다. feature 상의 임의의 지점 p_i와 다른 임의의 지점 p_j 사이의 attention 가중치는 다음과 같다.

$$W_d(p_i, p_j) = softmax(-\left\|Q(p_i) - Q(p_j)\right\|_2) \tag{3.2}$$

Q상에서의 거리가 가까울수록, attention 가중치는 크게 계산된다.

이러한 유사도 연산은 feature의 전체 영역이 아닌, Query가 되는 지점을 기준으로 십자 영역(criss-cross[10])에 대해서 연산 된다. 전체 영역이 아닌 십자 영역(criss-cross[10])에 대해서 연산을 수행할 때 두 가지의 이점이 있다. 첫 번째 장점은 모든 영역에 대해서 유사도 연산을 수행하지 않기 때문에, 연산량이 줄어든다. 이것은 criss-cross[10] 네트워크에서 제안하는 방식의 기본적인 장점이다. 또 다른 장점은 어떤 물체가 이미지에서 차지하는 공간은 상대적으로 크지 않다. 그렇기 때문에 한 물체와 관계된 픽셀을 찾아내기 위해서 이미지상의 모든 픽셀을 확인하는 것은 비효율적이다. 십자 영역(criss-cross[10])을 통해서 관계되는 모든 픽셀을 찾을 수는 없지만, 의미 있는 영역을 찾는 데는 충분하다.

1×1 Convolution layer가 인코더로부터 추출된 feature에 적용되어서 **V**(value)를 생성한다. 우리는 이 **V**(value)에 **Q**을 이용해서 만들어낸 attention 가중치를 이용해서 feature를 강화시킨다. 초기 feature인 $\mathbf{x}$을 강화된 feature 각각의 원소 단위로 덧셈 연산(Element-wise sum)을 수행한다. 이 과정을 통해서, RGB 기반의 feature $\mathbf{x}$는 깊이 context 정보가 간접적으로 융합되면서 강화된다. 이렇게 강화된 feature는 다음과 같이 정의된다.

$$\mathbf{y}_i = \sum\nolimits_{\forall j} W_d(p_i, p_j)\mathbf{V_j} + \mathbf{x_i} \tag{3.3}$$

요약하면, **DPA** 모듈은 RGB feature $\mathbf{x}$와 깊이 정보 $\mathbf{D}$를 입력으로 받는다. 깊이 정보 $\mathbf{D}$와 픽셀 위치정보 $\mathbf{P}$는 concatenation해서 Depth query $\mathbf{Q}$를 생성한다. attention 가중치 W_d는 $\mathbf{Q}$를 기반으로 계산되고, 이 가중치를 통해서 RGB feature를 강화시킨다. 최종적으로, 디코더를 통해서 깊이 정보를 통해서 강화된 feature가 들어가고, 픽셀의 클래스 레이블이 추론된다.

제 4 장 실 험 및 분 석

실험 환경에 대해서 소개하고, Cityscapes[3] 데이터 환경에서의 실험 결과를 소개한다. 실험은 기존의 RGB 기반의 네트워크에 **DPA** 모듈을 추가함을 통해서 수행했다. 실험의 기준이 된 RGB 기반의 네트워크는 총 3개이다. 실시간 작동에 중점을 둔 경량의 네트워크인 STDC[4]와 BiSeNetV2[24], 그리고 보다 무겁고 복잡하지만 성능에 중점을 두는 HRNet[18]이다. 결론적으로 **DPA** 모듈은 기준 네트워크의 종류와 관계없이 성능을 개선하였다.

4.1 실험환경

모델 : **DPA** 모듈의 효과를 확인하기 위해서, 총 3개의 네트워크를 사용했다. STDC[4]와 BiSeNetV2[24]는 경량화에 중점을 둔 성능이 우수한 네트워크이다. HRNet[18]은 보다 복잡하기만 강력한 성능에 더욱 중점을 둔 네트워크이다. 이 모델들은 디코더를 통해서 최종 결과를 추론하기 전 단계인 feature에 대해서 **DPA** 모듈을 적용한다. 기존 모델들이 사용하던 부가적인 손실함수 (예. boundary loss(STDC[4]), boost loss(BiSeNetV2[24]))는 유지하였다. 또한 STDC[4]와 BiSeNetV2[24]은 손실함수로 Cross-entropy 대신 OHEM[16]을 사용하기 때문에, 기존과 동일하게 OHEM을 적용하였다. HRNet[18]은 기존 네트워크와 동일하게 Cross-entropy를 사용하였다. 공정한 비교를 위해서, 모든 실험은 동일한 환경에서 수행했다.

실험 데이터 : Cityscapes[3]는 현존하는 데이터 중에서 야외 주행환경에 대해서 가장 포괄적인 정보를 가지고 있는 벤치마크 데이터이다. 또한 깊이 정보로 활용될 수 있는 disparity 정보를 제공한다. 이 데이터는 총 5,000장의 정교한 레이블 데이터를 제공한다. 총 레이블 클래스의 수는 19개이다. 5,000장 중

에서 2,975장은 훈련을 위한 것이고, 500장은 validation, 1525는 testing을 위한 데이터이다. RGB 이미지의 해상도는 2048×1024로 매우 고해상도이며, disparity 정보는 스테레오 카메라를 통해서 취득된 데이터로 다소 정교하지 못하다. 대략적으로 레이블링이 된 20k의 데이터도 제공하지만, 이 실험에서는 활용하지 않았다. disparity 정보를 바탕으로, 깊이 정보를 계산해서 학습에 활용했다.

구현방법: 실험 수행을 위해서 딥 러닝 프레임워크인 Pytorch[13]를 사용하였다. 1개의 Tesla A100 GPU를 사용했으며, CUDA 11.1, CUDNN 8.5.0, Pytorch 1.8.0.을 사용했다. BiSeNetV2[24]의 경우 batch의 크기는 16이며, STDC[4]는 48, HRNet[18]은 12를 적용했다. STDC[4]과 HRNet[18]은 기존의 네트워크가 사용했던 pretrained 모델로부터 학습을 시작했고, BiSeNetV2[24]은 무작위 초기화로 학습을 시작했다. 모델을 학습시키기 위해서 stochastic gradient descent (SGD)를 사용했으며, 0.9 momentum을 적용했다. Weight decay는 $5e^{-4}$를 적용했다. Learning rate는 "poly"라는 방식을 활용했으며, $(1 - \frac{iter}{iter_{max}})^{power}$ 이다. 이때 적용된 power는 0.9 이며, 초기 learning rate는 STDC[4], HRNet[18]의 경우 0.01이고, BiSeNetV2[24]은 0.005를 적용하였다. STDC[4]는 총 $60K$, BiSeNetV2[24]은 $150K$, HRNet[18]은 $120K$의 iteration의 학습이 수행되었다. 데이터의 증강을 위해서, 입력 이미지는 임의로 뒤집거나(Random flip), 비율이 조절되고(Random scale), 일부가 잘려져서(Random crop) 학습에 활용되었다. 학습 시에 사용된 입력 해상도는 1024×512로 잘려져서 사용되었다.

4.2 Ablation 분석

"깊이 및 픽셀 위치 기반 어텐션(Depth and Pixel-distance based Attention: **DPA**) 모듈"의 성능을 검증하고, 학습 모델의 query 형태를 결정하기 위해서

표 4.1: Depth query 형태에 따른 Cityscapes[3] validation set에서의 성능

Depth	Horizontal	Vertical	mIoU(%)
-	-	-	74.64
✓			76.07
✓	✓		75.90
✓		✓	75.61
✓	✓	✓	**76.50**

표 4.2: Depth query 형태에 따른 Scale Parameter 값

Depth query	α_d	β_x	γ_y
Depth	40.95	-	-
Depth + horizontal	30.05	7.07	-
Depth + vertical	41.69	-	0.33
Depth + horizontal + vertical	31.63	7.04	0.38

다양한 Depth query **Q**를 적용해서 실험을 수행했다. 실험은 STDC1-Seg75에서 수행했으며, 데이터는 Cityscapes[3] validation set에서 수행되었다. 신속한 실험 결과 확인을 위해서 batch의 크기는 48에서 24로 축소하여서 실행했다.

Depth query 형태에 따른 성능변화를 관찰하였다. **표 4.1**의 첫 번째 행은 **DPA** 모듈을 적용하지 않은 기본적인 STDC1-Seg75의 성능을 보여준다. 깊이 정보만 Depth query **Q**에 활용되었을 때는 성능이 1.4% 개선된다. 이 결과는 깊이 정보가 픽셀 간의 유사성을 계산하는 데 유용한 정보임을 보여준다. 깊이 정보와 수평 픽셀 위치 정보 및 수직 위치 정보가 concatenation 되면 0.43%의 추가 성능 향상이 있다. 픽셀 위치 정보, 즉 픽셀 사이의 거리가 제약으로 작용하여

깊이 값이 비슷하지만 멀리 떨어져 있는 값을 효과적으로 처리하고 성능을 향상시키는 것을 관찰할 수 있다. 그러나 수평 또는 수직 픽셀 위치 정보 중 하나만 concatenation 하면 성능 오히려 조금 하락했다. 일부 픽셀 위치정보만 제약조건으로 제공할 경우 깊이 정보 활용에 악영향을 미치는 것으로 확인됐다.

Scale Parameter는 학습이 가능하며, Depth query **Q**의 각 채널에 대한 가중치를 나타낸다. **표 4.2** 학습이 끝난 후 Scale parameter 값을 보여준다. 각 채널에 곱한 Scale parameter 증가하면 해당 채널의 작은 차이도 큰 차이로 계산된다. 따라서 Scale parameter 값이 클수록 국소 영역에 더 많은 attention이 집중되는 효과가 나타난다. α_d, β_x 및 γ_y는 각각 깊이, 수평 및 수직에 적용되는 값이다. 값의 수렴지점은 **Q**의 형태에 따라서 다르지만, β_x는 약 7로 수렴하고 γ_y는 약 0.3으로 수렴한다. 이는 수평 영역에 대한 attention이 상대적으로 국지적 영역으로 제한되고 수직 영역에 대해서는 상대적으로 넓은 영역이 attention 된다는 것을 의미한다. 이는 이미지의 특성을 고려한 자연스러운 결과이다. 수평 위치에서는 유사한 깊이를 가진 다른 물체가 있을 가능성이 높기 때문에 attention 영역이 국소적으로 제한된다. 수직 위치에서는 비슷한 깊이를 가진 물체가 존재하기 어렵기 때문에 상대적으로 전체 영역으로 확장된다. 또한 수평 위치 정보가 깊이와 concatenation 되면 α_d가 상대적으로 작은 값으로 수렴된다는 것을 관찰할 수 있다. 이는 수평 위치 정보가 국소적으로 attention 영역을 제한하기 때문에 깊이 값 측면에서 더 전역적으로 바라보더라도 비슷한 깊이를 가진 다른 물체의 영향을 덜 받기 때문이다. 하지만 수직 위치 정보와 깊이가 concatenation 되면 깊이 값에 대한 Scale parameter α_d가 더 큰 값으로 수렴된다. 이는 수평 영역에 대해 제약이 없기 때문에 수평 영역에서 발생하는 다른 물체의 영향을 줄이기 위해 다소 큰 값으로 수렴한 것으로 판단된다. 이 결과로부터 Scale parameter가 **Q**의 채널 간 스케일을 적절하게 조정함을 알 수 있다.

표 4.3: Cityscapes[3] Validation 데이터에서의 클래스별 IoU(%) 성능

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
BiSeNetV2[24]	97.9	83.0	91.8	46.5	56.1	61.7	68.7	77.3	92.1	63.0	**94.6**	80.8	57.9	94.5	**68.9**	78.8	**75.0**	58.8	76.8	75.0
BiSeNetV2[24] + DPA	**98.1**	**84.3**	**92.3**	**54.3**	**60.1**	**64.6**	**70.3**	**79.1**	**92.3**	**64.2**	94.4	**82.0**	**60.0**	**94.7**	66.2	**80.6**	73.5	**61.0**	**77.0**	**76.3**
STDC1-Seg75[4]	**98.1**	**84.2**	91.7	49.8	58.0	58.1	66.0	75.4	91.6	61.2	94.3	79.3	58.9	94.4	**74.9**	81.4	65.1	58.1	75.2	74.5
STDC1-Seg75[4] + DPA	98.0	83.9	**92.0**	**55.0**	**58.4**	**60.8**	**68.7**	**77.1**	**91.8**	**61.4**	**94.6**	**79.7**	**60.1**	**94.5**	74.3	**83.8**	**71.4**	**62.0**	**75.3**	**75.9**
STDC2-Seg75[4]	98.2	**85.3**	92.3	56.0	59.1	60.7	69.3	**78.0**	91.9	62.3	94.6	80.3	60.3	95.0	**81.2**	**87.8**	75.1	60.3	76.0	77.0
STDC2-Seg75[4] + DPA	98.2	85.1	**92.5**	**60.0**	**60.3**	**61.9**	**70.4**	77.8	**92.0**	**64.1**	94.6	**80.5**	**60.5**	95.0	81.1	85.0	**76.5**	**64.3**	**76.4**	**77.7**
HRNet[18]	**98.5**	**87.1**	93.5	58.6	64.2	**71.2**	**75.1**	82.0	**93.2**	65.5	**95.2**	84.8	**66.4**	**95.7**	79.5	91.1	83.3	**70.0**	**80.4**	80.8
HRNet[18] + DPA	98.4	86.9	**93.6**	**62.5**	**66.7**	71.1	74.9	**83.0**	93.1	**65.5**	94.9	84.7	66.6	95.7	**84.4**	**92.0**	**85.7**	68.8	79.1	**81.4**

표 4.4: Cityscapes[3] Test 데이터에서의 클래스별 IoU(%) 성능

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
BiSeNetV2[24]	98.3	83.6	91.7	44.8	52.5	59.8	70.7	74.4	92.8	69.3	94.7	84.0	65.2	95.1	56.3	73.3	60.1	58.1	73.2	73.6
BiSeNetV2[24] + DPA	98.3	**84.5**	**92.1**	**49.2**	**55.4**	**61.9**	**71.6**	**75.6**	**93.0**	**71.1**	**94.8**	**84.8**	**67.7**	**95.2**	**62.9**	**74.3**	**69.5**	**61.8**	**74.5**	**75.7**
STDC1-Seg75[4]	98.5	85.2	91.8	51.1	51.7	58.3	68.2	**73.3**	92.7	70.6	94.8	82.6	66.5	95.1	68.0	79.3	70.4	60.5	71.2	75.3
STDC1-Seg75[4] + DPA	98.5	**85.4**	**92.2**	**54.0**	**53.8**	**59.4**	70.2	74.1	92.7	**71.0**	**94.9**	**83.4**	**67.3**	**95.3**	**70.7**	**83.8**	**80.1**	**62.7**	**72.4**	**76.9**
STDC2-Seg75[4]	98.5	85.4	92.3	**54.6**	56.0	60.0	70.3	74.2	92.8	70.6	94.7	83.6	67.7	95.5	**71.4**	81.1	74.9	63.7	72.8	76.8
STDC2-Seg75[4] + DPA	**98.6**	**85.8**	**92.4**	50.7	**56.5**	**61.1**	**71.5**	**74.8**	92.8	**70.7**	**95.0**	**84.1**	**69.2**	**95.6**	71.2	**83.3**	**79.5**	**65.8**	**72.8**	**77.5**
HRNet[18]	98.7	86.9	93.2	49.1	61.6	**71.1**	**78.4**	**81.4**	93.8	71.3	95.7	**87.9**	**73.7**	96.0	71.4	80.1	74.3	**72.3**	78.0	79.7
HRNet[18] + DPA	98.7	**87.1**	**93.4**	**54.7**	**61.9**	70.4	77.9	80.6	**93.9**	**72.9**	**95.8**	87.6	73.2	**96.2**	71.4	**86.3**	**80.1**	71.6	**78.3**	**80.6**

4.3 Quantitative 실험결과

Training 데이터에서 학습을 수행하고, Validation 데이터에서 성능을 확인하고, Training과 Validation을 모두 활용해서 Test 데이터에서 성능을 확인했다. **DPA** 모듈의 정확한 성능을 확인하기 위해서 추가적인 "평가 기술(Evaluation technique)"은 사용하지 않았다.(예. multi-scale testing, flipping.) 첫 번째로, **표 4.3, 4.4**는 "baseline 모델 + **DPA**"을 적용했을 경우를 보여주고 있다. Segmentation의 성능은 baseline 모델의 종류와 관계없이 개선되었다. BiSeNetV2[24]의 경우, Validation 데이터에서 mIOU가 1.3% 개선되었다. 또한 대부분의 클래스에 대해서 성능이 향상되었다. 특히 **wall**, **fence**, **pole**와 같은 클래스의 성능 향상이 두드러짐을 알 수 있다. 이는 기존의 RGB 기반 segmentation 과정에서 다른 물체와 패턴이 유사해 잘 감지되지 않는 물체를 깊이 정보를 통해 효과적으로 감지할 수 있음을 보여준다. 이러한 특성은 Test 데이터에서도 유사한 경향을 보여준다. 결과적으로 Test 데이터에서 성능이 2.1% 향상된다. STDC[4]는 작은 모델인 STDC1과 크고 복잡한 모델인 STDC2가 있다. 모델의 크기에 상관없이 성능이 개선되었다. 성능은 Validation 데이터에서 STDC1과 2에서 각각 1.4%와 0.7% 향상된다. 이전의 BiSeNetV2[24]과 유사하게, 대부분의 클래스에서 성능이 향상되었다. Test 데이터에서 Validation 데이터와 유사하게 각각 1.6%와 0.7%의 성능 개선이 있었다. HRNet[18]의 경우, Validation 데이터에서 0.6% Test 데이터에서는 0.9%의 개선이 있었다. 모델의 복잡성과 크기가 커질수록 성능 향상 정도는 감소하는 경향이 있었다. 스테레오 카메라를 기반으로 한 깊이 정보의 정확도의 한계로 인해 기준 모델의 성능이 증가함에 따라 성능 향상 정도가 감소한다고 추정된다.

표 4.5: Cityscapes[3] Test 데이터에서 네트워크 복잡성에 따른 효율성 비교

	Params/M	FLOPs/G	mUoU(%)
STDC1	12.1	57.7	75.3
STDC2	16.1(33%↑)	87.5(52%↑)	76.8
STDC1+ DPA	12.3(2%↑)	59.7(4%↑)	76.9
STDC2 + DPA	16.3(35%↑)	89.5(55%↑)	77.5

표 4.6: Cityscapes[3] Test 데이터에서 병렬인코더 사용 네트워크와의 효율성 비교

	Params/M	FLOPs/G	mUoU(%)
ESA(RGB)[15]	32.1	54.2	72.9
ESA(RGBD)[15]	46.9(46%↑)	87.3(60%↑)	**75.7**
STDC2	16.1	87.5	76.8
STDC2 + DPA	16.3(1.2%↑)	89.5(2.3%↑)	**77.5**

4.4 효율성 분석

제안하는 **DPA** 모듈이 효율성 관점에서 도움이 되는지 확인하기 위해 STDC[4]에서 비교를 수행한다. STDC[4]는 네트워크의 복잡성에 따라 두 가지 유형의 인코더를 제공한다. STDC2는 STDC1에 비해 더 깊은 네트워크로 성능을 향상시키는 데 사용된다. **표 4.5**에서, STDC2는 Parameter 수와 FLOPs를 모두 크게 증가시키는 것을 확인할 수 있다. STDC2를 STDC1+DPA와 비교하였을 때, DPA에 의한 Parameter 및 FLOPs의 증가가 STDC2에 의한 증가량 비해 매우 적은 것을 알 수 있다. 그러나 STDC1 + DPA의 성능이 0.1% 더 높은 것을 확인할 수 있다. 이것은 **DPA** 모듈이 깊이 정보를 활용하여 성능을 개선할 뿐만 아니라 효율성의 관점에서도 이점이 있는 것을 확인할 수 있다. **표 4.6**은 병렬 인코더를 통해서 깊이 정보를 활용하는 방법과 **DPA** 모듈을 사용하는 방식과의 효율성 차이를 보여준다. ESA[15]는 RGB와 깊이 정보를 각각의 인코더를 적용하는 병렬형 네트워크이다. 두 개의 인코더를 사용하기 때문에 ResNet34[7]과 같은 얕은 네트워크를 활용함에도 불구하고 깊이 정보 처리를 위한 인코더를 사용하면 FLOPs와 Parameter가 크게 증가한다. Parameter는 46% 증가하고 FLOPs는 60% 증가한다. 반면, **DPA** 모듈이 STDC2에 적용된 경우, 각각 1.2%와 2.3% 증가한다. 이것은 **DPA** 모듈의 사용이 두 개의 병렬형 인코더를 사용하는 것보다 효율적인 방법임을 보여준다.

4.5 Qualitative 실험결과

그림 4.1은 baseline 네트워크와 **DPA** 모듈의 적용 시의 Semantic segmentation 결과를 보여준다. 첫 번째 행은 BiSeNetV2[24]에 **DPA** 모듈을 적용했을 때의 결과이다. 첫 번째 줄의 왼쪽 트럭은 이미지의 큰 부분을 차지하고 있고 일부가 잘려져 있다. RGB 기반 네트워크는 제한된 네트워크의 Receptive field와 모호성으로 인해 일부 부분이 잘못 추론되는 결과를 초래했다. **DPA** 모듈

을 적용 시 깊이 정보를 통해 잘못 추론된 부분도 트럭의 일부임을 유추하고, 그 결과 오류가 수정되었다. 두 번째 줄에서, 왼쪽 부분의 차량은 매우 적은 부분만 이미지에 나와 있어서 어떤 물체인지 매우 판단하기 어렵다. RGB만 사용한 방식에 비해서, **DPA** 모듈은 추론에 오류가 발생한 부분을 많이 개선하였다. 두 번째 행과 세 번째 행은 STDC[4]에 대한 결과를 보여준다. 두 번째 행의 STDC1-Seg75[4] 노란 상자 안을 보면, 빛의 부족이나 배경의 복잡성으로 인해 잘못 추론된 부분이 효과적으로 개선되는 것을 관찰할 수 있다. 하지만 두 번째 줄의 왼쪽 영역의 풀과 인도의 경우 기본 STDC[4]가 보다 나은 성능을 보여주는 것을 확인할 수 있다. 이것은 잔디 부분과 보도 사이의 깊이 값의 현저한 차이가 없기 때문에 깊이 정보를 통한 개선이 없는 것으로 추정한다. 세 번째 행은 STDC2-Seg75[4]의 결과이다. 성능 개선을 위해서 보다 깊은 인코더를 사용한다. 여기서도 물체의 잘림이나 빛의 부족 등에 의한 추론에 오류가 발생한 영역을 개선하는 것을 관찰할 수 있다. 마지막 행은 HRNet[18]의 결과를 보여준다. 깊이 정보를 통해 하늘과 건물을 구분하지 못하는 부분이 잘 구분된 것을 알 수 있다. 기준 네트워크의 성능이 향상되면서 추론성능의 질적 차이는 다소 감소하지만, 잘린 물체나 빛의 부족은 여전히 어려움을 야기하며 **DPA**모듈은 이를 효과적으로 개선하는 것을 확인할 수 있다.

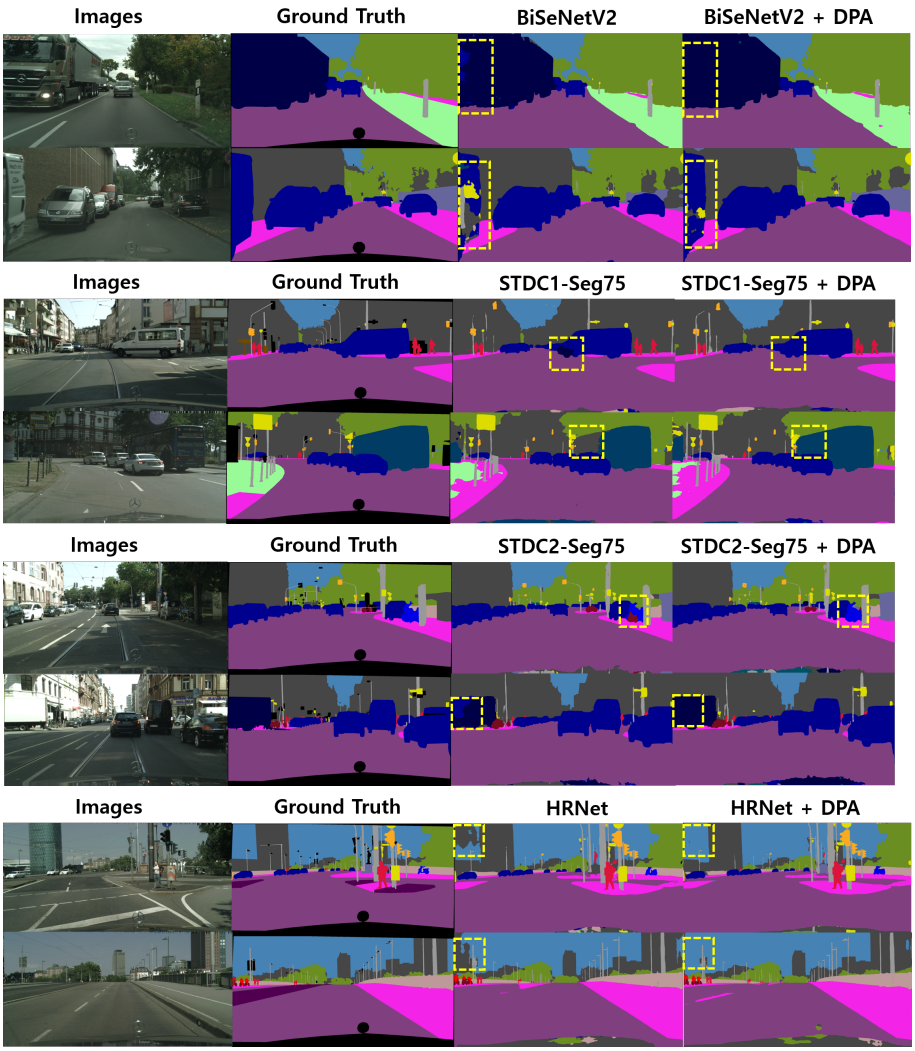


그림 4.1: Cityscapes[3] Validation 데이터에서의 Qualitative 결과 예시

제 5 장 결 론

본 논문에서는 깊이 및 픽셀 위치 기반 어텐션(**DPA**) 모듈을 제안한다. 픽셀 간의 유사성은 깊이 정보와 픽셀 위치를 사용하여 계산된다. 계산된 유사성을 기반으로 Attention이 필요한 영역을 Attention 가중치 형태로 표현하고, 이를 통해 RGB 기반 feature를 깊이 정보가 포함된 feature로 증강한다. Scale Parameter는 서로 다른 벡터 공간의 축을 깊이와 픽셀 위치에 효율적으로 정렬하여 픽셀 간의 상관관계를 계산한다. 실험 결과 네트워크의 구조를 크게 변경하지 않고 모듈을 추가하여 깊이 정보를 사용이 가능하고, 기존 RGB 기반 Segmentation 네트워크의 성능을 향상시킬 수 있음을 확인하였다. 향후 깊이 정보를 기반으로 픽셀 간의 상관관계를 표현하는 그래프를 형성하는 방식을 통해서 Segmentation의 성능을 개선하는 방법을 모색하고자 한다.

참고 문헌

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[2] Yunlu Chen, Thomas Mensink, and Efstratios Gavves. 3d neighborhood convolution: Learning depth-aware features for rgb-d and rgb semantic segmentation. In *2019 International Conference on 3D Vision (3DV)*, pages 173–182. IEEE, 2019.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9716–9725, 2021.

[5] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019.

[6] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1440–1444. IEEE, 2019.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[10] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[11] Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*, 2018.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Daniel Seichter, Mona Köhler, Benjamin Lewandowski, Tim Wengefeld, and Horst-Michael Gross. Efficient rgb-d semantic segmentation for indoor scene analysis. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13525–13531. IEEE, 2021.

[16] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE*

transactions on pattern analysis and machine intelligence, 43(10):3349–3364, 2020.

[19] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[20] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[21] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[22] Yajie Xing, Jingbo Wang, and Gang Zeng. Malleable 2.5 d convolution: Learning receptive fields along the depth-axis for rgb-d scene parsing. In *European Conference on Computer Vision*, pages 555–571. Springer, 2020.

[23] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3684–3692, 2018.

[24] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021.

[25] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[26] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pages 173–190. Springer, 2020.

[27] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[28] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 548–557, 2019.

[29] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

ABSTRACT

Semantic segmentation is the most comprehensive way to understand images. It is to assign a semantic class label to every pixel in the image. It is a very important technology because it corresponds to recognition from the perspective of operating autonomous driving and robots. Recently, attempts have been made to use depth information to improve the performance of semantic segmentation. However, most attempts have been made in indoor environments rather than outdoors. There are also reasons that make depth information difficult to exploit easily. First, it is difficult to obtain accurate and dense depth information in an outdoor environment. Second, when processing depth information as input to a network, an additional encoder is required, so a new network design is required.

In this paper, we overcome the difficulties and find ways to use depth information efficiently. To this end, we propose a novel **D**epth and **P**ixel-distance based **A**ttention (**DPA**) module. This module utilizes depth information and uses it to infer correlations between pixels. It is computed using the fact that pixels belonging to the same object have similar depth values. It is robust to the accuracy of depth information because only relative differences in depth are used. In addition, **DPA** is a simple plug-in module that can be easily exploited to existing RGB segmentation networks. It does not require a new network design for depth information processing and can be easily applied to existing RGB-based networks that work well. It is also much more efficient from a computational point of view, since no additional encoder is required to process the depth information. **DPA** does not use depth information as input, but indirectly provides

depth information to RGB-based features. Through this, the RGB-based feature is augmented.

Performance and efficiency are verified by applying the **DPA** module to various baseline networks. Regardless of the type of baseline model, we improved the performance of semantic segmentation and verified that it is more efficient in terms of the amount of computation compared to the existing method of using depth information as an input.

keywords: Semantic Segmentation, Attention, Neural Network, Deep Learning
Student Number: 2021-21575