Auto White Balance
for multi-illuminant scene

JYP

팀원: 박윤정 유수민 조윤수

지도교수: 김선주

조교: 김동영

목차

1. 연구 주제

다중 조명 상황에서의 Auto White Balance 구현:

Auto White Balance(이하 AWB)는 사진에서 조명의 영향을 제거해 물체의 고유 색상이 잘 표현되도록 하는 기술이다. 장면 내에 여러 개의 조명이 존재하는 경우 적용 가능한 AWB 알고리즘에 대하여 연구하고자 한다.

2. 연구의 필요성

2.1 기존 연구의 한계점

기존의 AWB 알고리즘은 카메라 센서가 장면의 광원 및 조도를 추정한 후 이를 기반으로 보정을 하는 방식으로, 추정에 있어서 단일 광원을 가정한다. 단일 광원 AWB 알고리즘으로는 Gray-world methods, white-patch hypothesis와 같은 Statistical methods와 gamut-based methods와 같은 Learning-based methods가 있다. 그러나 현실 상황에서는 하나의 장면 내에 다양한 광원이 존재하기에, 이러한 기존 AWB 알고리즘을 적용할 경우 빨갛거나 파랗게 물드는 등 실제 색과 전혀 다른 결과물을 낼 가능성이 매우 높아진다[1].

2.2 Large Scale Multi-Illuminant Dataset

관련 연구로 제시된 'Large Scale Multi-Illuminant (LSMI) Dataset for Developing White Balance Algorithm under Mixed Illumination'은 dataset의 각 이미지에 대하여, 조명의 종류, 광원의 색도, 광원의 혼합 비율을 픽셀 단위로 제시한다[2]. 더불어, LSMI dataset에 CNN 기반의 pixel-level White Balance 알고리즘을 적용하여 pixel-level이 patch-based method보다 성능이 뛰어남을 보여주었다. CNN 기반 모델로는 U-net과 HDRnet이라는 두 기본적인 모델을 사용하였는데, 해당 연구에서는 사용한 CNN 기반 모델에 있어 여전히 개선의 여지가 있음을 시사하였다.

3. 연구 내용

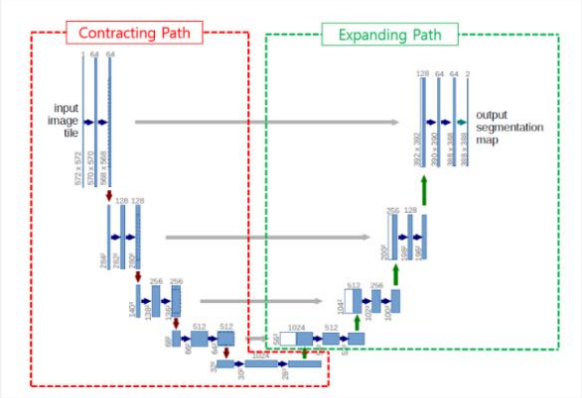
3.1 연구 주제 및 목표 선정

LSMI dataset을 사용하며 개선된 AWB 모델을 찾는다. 기존 모델을 수정하여 U-net3+ 모델을 적용시켜볼 뿐만 아니라, vision transformer를 활용하여 심화 모델을 구현한다.

3.2 기존 알고리즘 분석

LSMI dataset 연구의 dataset과 모델 구조를 이해하기 위해, 해당 연구에서 제시된 코드를 활용하였다. U-net 모델을 학습, 테스트하며 모델의 구조 및 코드의 구성을 이해하였다[3].

U-net은 의료 이미지 분야에서 image segmentation을 목적으로 제안된 모델로, End-to-end 방식의 fully convolutional network를 기반으로 하는 모델이다[4]. 입력 이미지의 전반적인 컨텍스트 정보를 얻기 위한 Contracting path와, feature map을 upsampling하고 컨텍스트 정보와 결합하여 정확한 localization을 하는 Expanding path가 대칭 형태로 구성되어 있다.

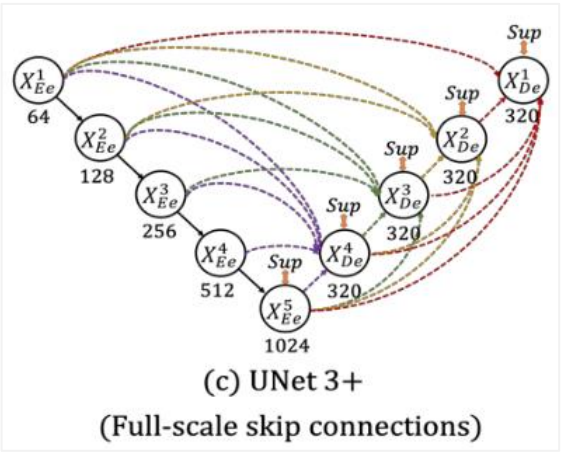


Structure of U-Net[4]

이 연구에서의 AWB는 이미지의 illuminants의 chrominance 값인 u와 v를 예측하는 것으로, 연속적인 값을 입력 받아 연속적인 값을 예측하는 것이기 때문에 regression task에 속한다. 그럼에도 불구하고 semantic segmentation task를 위한 모델인 U-net을 사용한 이유는, 해당 모델이 픽셀 별 예측을 하기 때문이다. Segmentation task는 이미지의 영역 또는 부분을 해당 클래스 레이블로 그룹화하기 위하여 이미지를 픽셀 단위로 구분해 픽셀 별 classification을 수행하는 작업이다. 해당 연구에서는 하나의 이미지 내에서 u와 v를 픽셀별로 예측하고자 하기에, input image와 같은 크기의 output map에 있어 u, v값을 pixel별로 예측할 수 있도록 segmentation task의 모델을 수정하여 적용하였다.

3.3 U-net3+

U-net3+는 원래 의료 영상을 다루기 위해 개발된 encoder-decoder 구조의 모델이다[5]. image segmentation 문제에서 우수한 성능을 보이며, 오늘날 image processing, image-to-image translation과 같은 pixel-level의 task에서 널리 사용된다.



Structure of U-Net3+[5]

U-net3+가 기존 u-net과 다른 점은 다음과 같다.

- Full-scale skip connection: low scale과 high scale의 feature를 합쳐 더 적은 parameter로 더 정확한 segmentation을 얻는다.
- Full-scale deep supervision: 각각의 decoder에서의 결과를 deep supervision으로 사용

하여 성능을 개선한다.

- Classification Guided Module: CGM을 사용하여 over-segmentation을 방지하고 더 정확한 결과를 낸다.

3.4 U-net 계열 모델로 LSMI dataset 학습

다중 조명 장면의 경우 픽셀 별 예측이 patch-based 예측보다 더 좋은 성능을 보였다는 기존 논문을 참고하여, Unet3+를 사용하여 연구를 진행하였다. LSMI dataset에서 사용하였던 기존 U-net 모델을 U-net3+ 모델로 수정하기 위하여 official U-net3+의 class UNet_3Plus를 사용하였다[6]. U-net3+는 본래 semantic segmentation이 목적인 모델이기 때문에 코드의 구조를 일부 수정하였다. 원래 모델의 output인 segmentation map에는 각 픽셀마다 class일 확률이 처음에 저장되는데, regression task에 맞게 각 픽셀마다 예측된 u, v값이 저장되도록 output layer를 수정하였다. 모델의 arguments와 train하는 부분 또한 수정하였다.

U-net3+의 모델을 처음 학습시킬 때, 모델의 크기가 매우 커 batch size를 8로 진행하였다. 테스트 결과 MAE(Mean Angular Error)의 최대값은 약 2도 개선되었으나, 평균값과 중앙값은 소수점 한 자리 차이로 U-net보다 높게 나와 기존 모델에 비해 뚜렷한 성능 개선이 있다고 보기 어려웠다. 이에 U-net 계열 모델의 batch size, learning rate, learning rate decay, initial weight를 다양하게 변경해보며, 모델과 task에 맞는 최적의 값을 찾기 위한 실험을 진행하였다.

다음은 U-net 계열 모델로 실험을 진행한 내용이다.

[U-net]

1) pre-trained 모델 사용

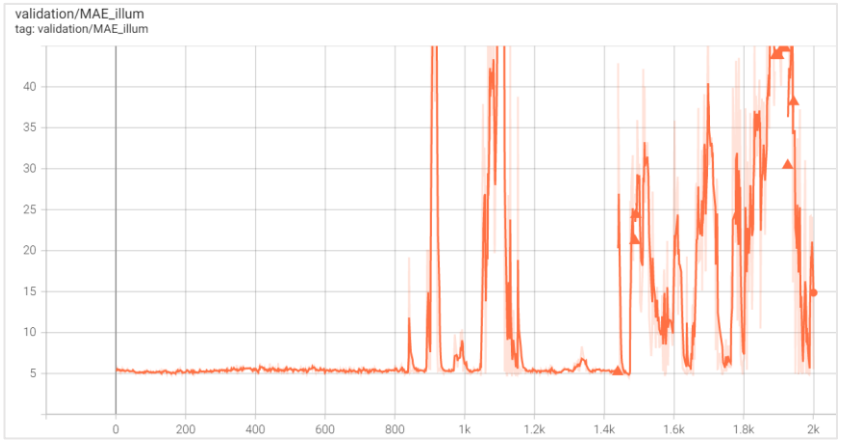
- batch size = 32

- test 결과:

	Mean	Median	Max
Loss	0.6750193232158199	0.015551890712231398	12.747682571411133
Pred_loss	0.6750193232158199	0.015551890712231398	12.747682571411133
MAE_illum	3.084842178821564	2.042070984840393	15.192743301391602
MAE_rgb	3.175431336641312	2.390289306640625	13.562090873718262
PSNR	40.82068655395508	41.10082244873047	58.39793395996094

2) batch size 8로 train 진행

- 2000 epoch까지 진행하였으나 학습이 제대로 되지 않았음.



- test 결과: 1)보다 확연히 낮은 성능

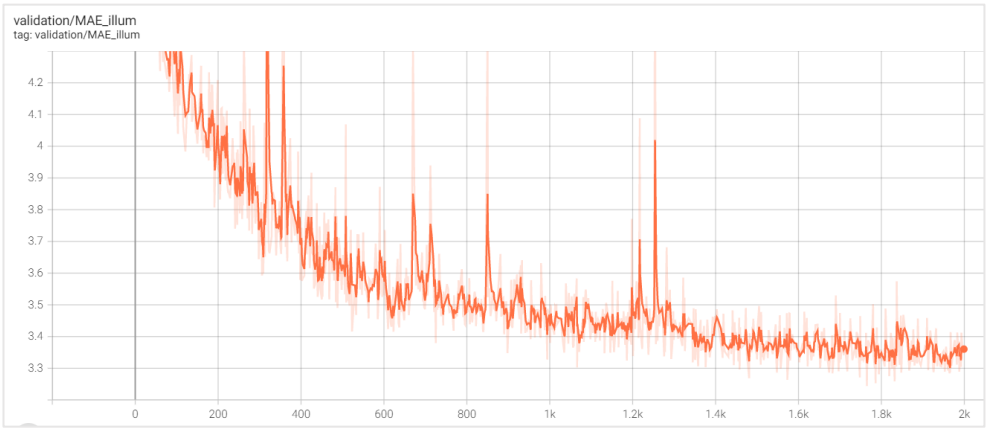
	Mean	Median	Max
Loss	0.6556932591963559	0.03301318176090717	12.537454605102539
Pred_loss	0.6556932591963559	0.03301318176090717	12.537454605102539
MAE_illum	4.27299546289444	3.498018264770508	17.304292678833008
MAE_rgb	4.51566597032547	3.948768138885498	18.065723419189453
PSNR	37.45086754608154	37.25107002258301	58.804588317871094

[U-net3+]

(기본적으로 모두 batch size 8로 진행)

1) batch size 8로 train 진행

- 2000 epoch까지 진행



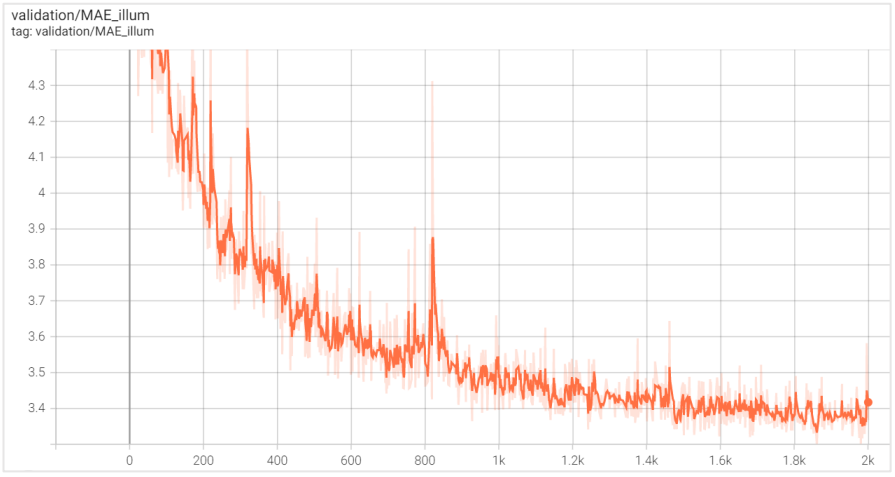
- test 결과: max값만 약 2도 개선

	Mean	Median	Max
Loss	0.6757568385833874	0.018872796557843685	12.987634658813477
Pred_loss	0.6757568385833874	0.018872796557843685	12.987634658813477
MAE_illum	3.223956574201584	2.4187151193618774	13.881534576416016
MAE_rgb	3.383839700222015	2.880157709121704	14.299846649169922
PSNR	39.79251309204101	40.01227569580078	58.060157775878906

2) lr_decay 변경

- 1200 → 800

- 2000 epoch까지 진행

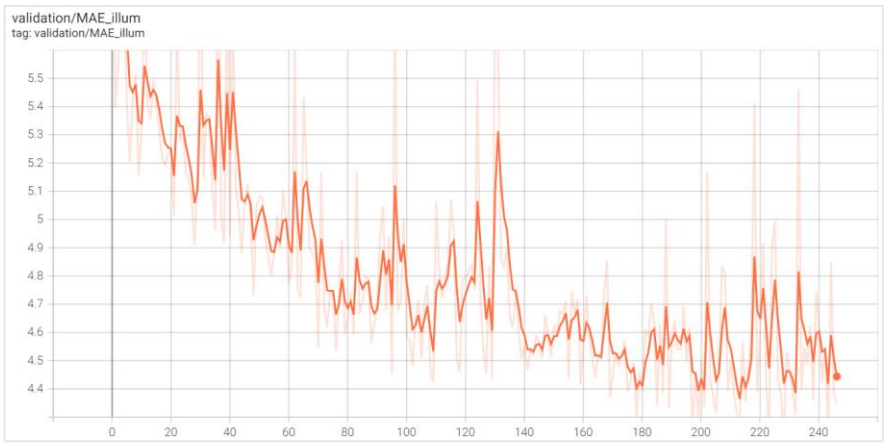


- test 결과: 성능 개선 없음.

	Mean	Median	Max
Loss	0.686239316022722	0.018876037560403347	13.068746566772461
Pred_loss	0.686239316022722	0.018876037560403347	13.068746566772461
MAE_illum	3.2258505873680114	2.4601714611053467	14.466060638427734
MAE_rgb	3.384591878414154	2.97456431388855	14.383424758911133
PSNR	39.99967610931397	40.415775299072266	58.1892204284668

3) lr 변경

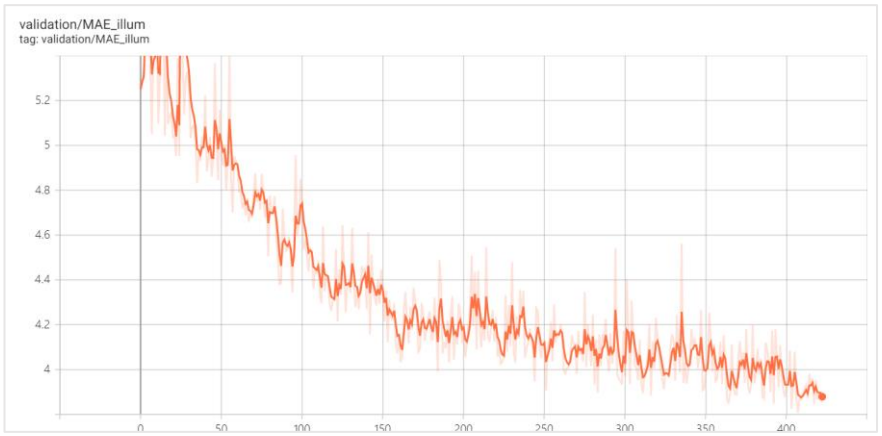
- 5e-4 → 5e-3



- 성능이 좋지 않아 학습 중단.

4) lr 변경

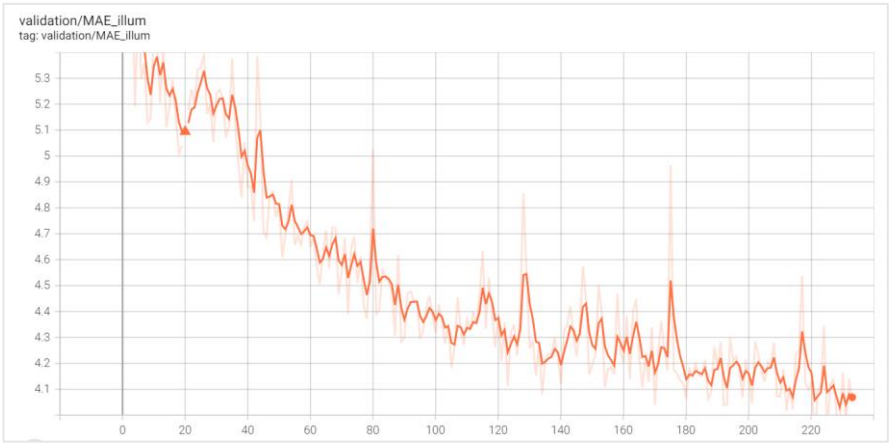
- 5e-4 → 1e-3



- 성능이 좋지 않아 학습 중단.

5) lr 변경

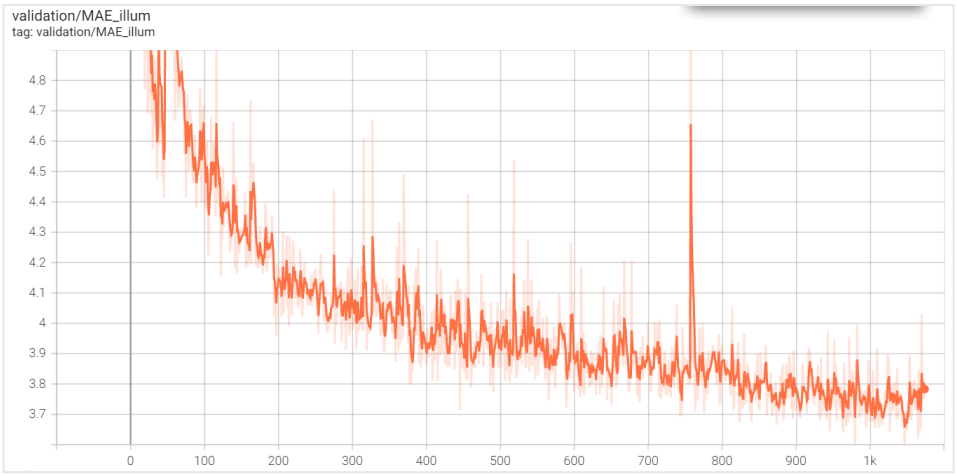
- 5e-4 → 9e-4



- 성능이 좋지 않아 학습 중단.

6) lr 변경

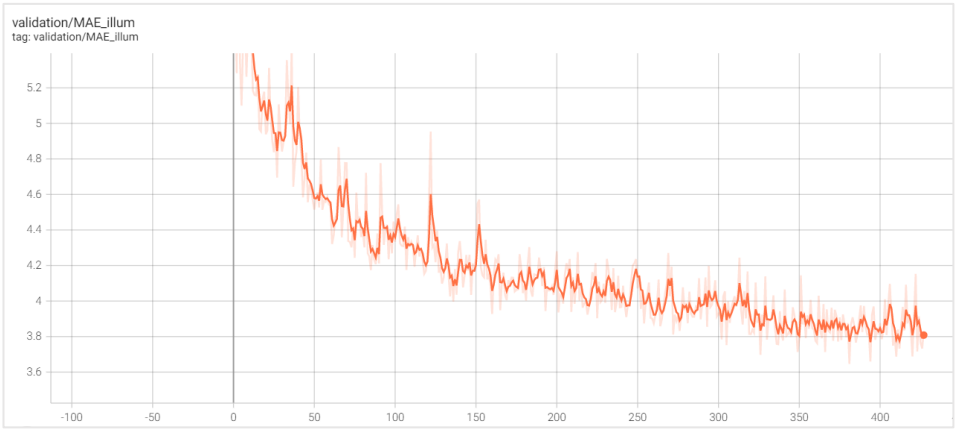
- 5e-4 → 6e-4



- 성능이 좋지 않아 학습 중단.

7) Beta1 변경

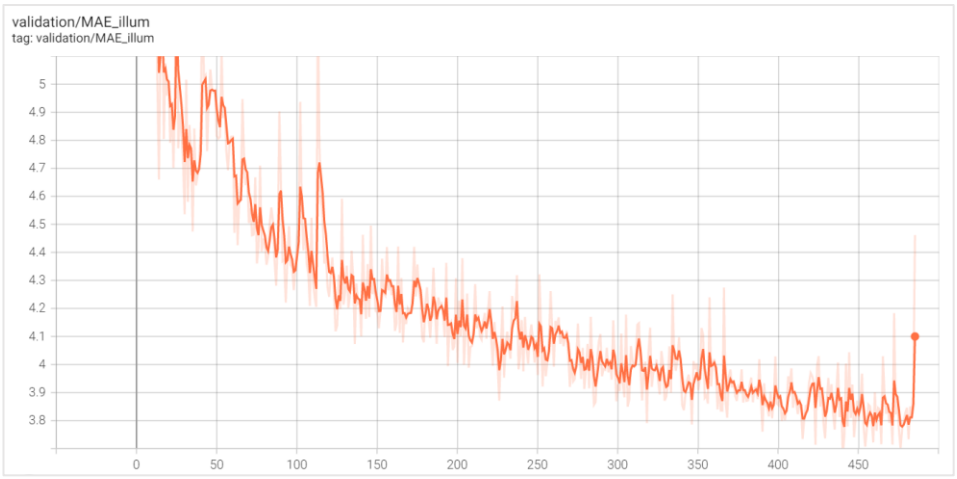
- 0.5 → 0.9



- 성능이 좋지 않아 학습 중단.

8) Beta1, lr 변경

- Beta1: 0.5 → 0.9, lr: 5e-4 → 1e-3



- 성능이 좋지 않아 학습 중단.

9) batch size 변경

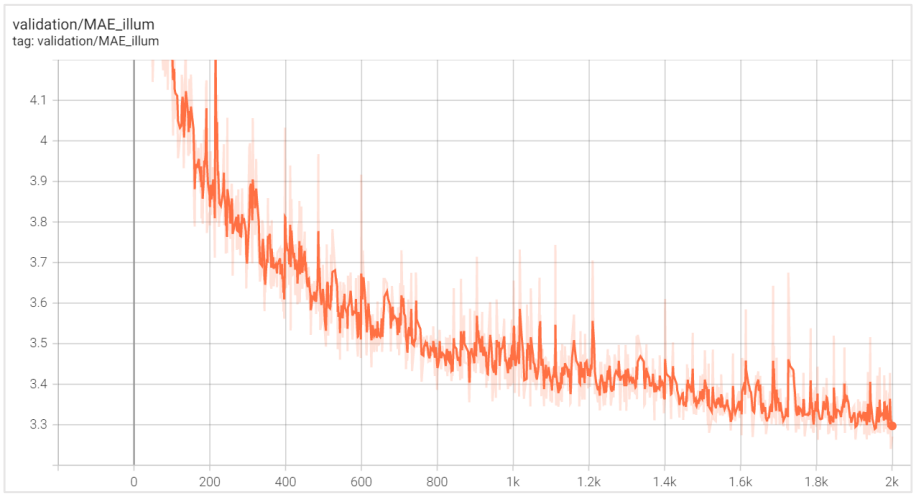
- 8 → 10

- 메모리 부족으로 학습이 중단됨.

10) initial_weight 변경

- kaiming → 파이토치 기본

- 2000 epoch까지 진행



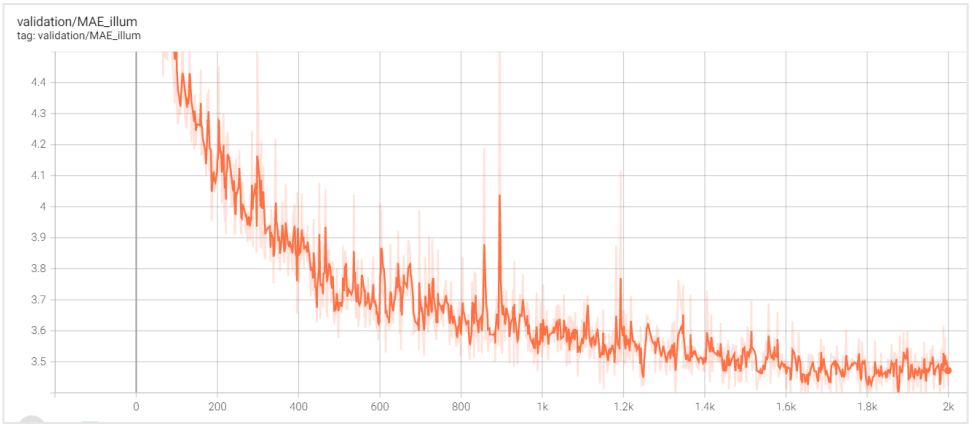
- test 결과: U-net3+ 1)보다는 개선되었지만, U-net보다는 좋지 않았음.

	Mean	Median	Max
Loss	0.6802572672951501	0.019749599508941174	12.992382049560547
Pred_loss	0.6802572672951501	0.019749599508941174	12.992382049560547
MAE_illum	3.1937566933631896	2.4011659622192383	13.74806022644043
MAE_rgb	3.3469357435703277	2.842648983001709	14.092072486877441
PSNR	40.04792473602295	40.51704978942871	58.01219940185547

11) initial_weight, lr 변경

- initial_weight: kaiming → 파이토치 기본, lr: 5e-4 → 55e-5

- 2000 epoch까지 진행



- test 결과:

	Mean	Median	Max
Loss	0.680783356401138	0.021128499880433083	13.02160930633545
Pred_loss	0.680783356401138	0.021128499880433083	13.02160930633545
MAE_illum	3.2838936536312104	2.410258650779724	13.966195106506348
MAE_rgb	3.436760799884796	2.86603844165802	14.44478702545166
PSNR	39.91862512207031	40.22232627868652	57.504150390625

위 결과 중 2000 epoch만큼 학습이 진행된 모델들의 성능을 비교하였다. MAE(Mean Angular Error)의 평균값을 성능 평가 지표로 사용하였으며, 아래는 그 수치가 낮은 순서대로 나타낸 표이다. MAE 값은 작을수록 AWB 적용 이미지가 groundtruth 이미지와 유사하며 성능이 좋음을 의미한다.

	모델	Batch size	Learning rate	Learning rate decay	Initial weight	MAE_illum mean
1	U-net	32	0.0005	1200	파이토치 기본	3.0848
2	U-net3+	8	0.0005	1200	파이토치 기본	3.1938
3	U-net3+	8	0.0005	1200	Kaiming	3.2240
4	U-net3+	8	0.0005	800	Kaiming	3.2259
5	U-net3+	8	0.00055	1200	파이토치 기본	3.2839
6	U-net	8	0.0005	1200	Kaiming	4.2730

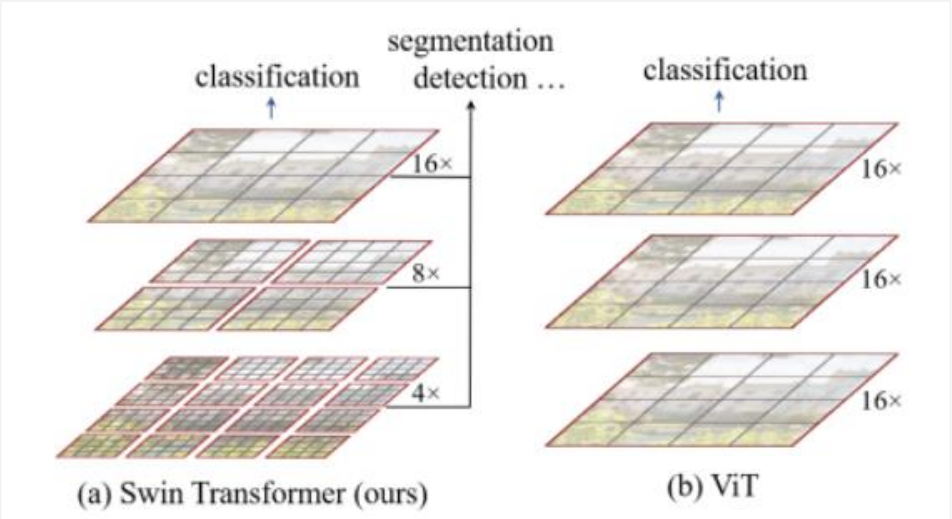
3.5 SWIN transformer + UperNet

Encoder-Decoder 구조를 사용하였다. Backbone으로는 SWIN transformer를 사용해 feature를 추출하고, decode_head로는 UperNet을 사용해 upsampling을 하였다[7]. Backbone network model은 classification용으로 만든 모델의 FC 부분을 수정하여 다른 용도로 사용할 수 있도록 한다.

3.5.1 Swin transformer

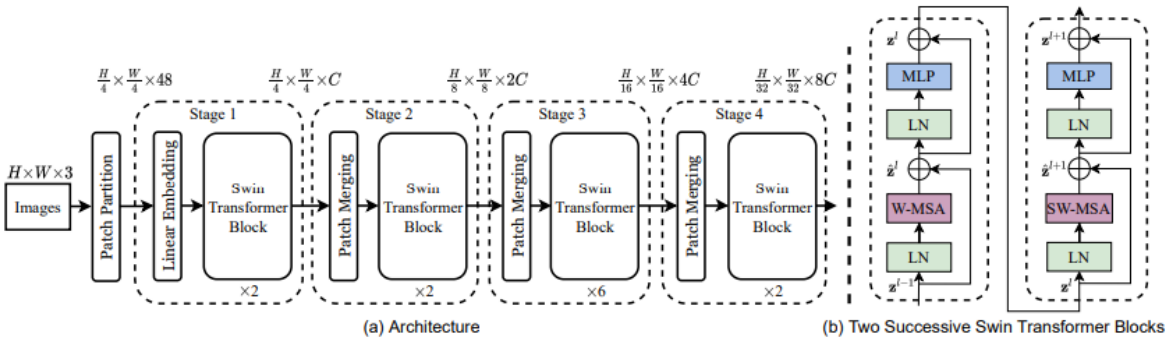
“Hierarchical Vision Transformer using Shifted Windows” 논문에서 소개된 모델로, Computer Vision분야에서 general-backbone으로 사용될 수 있는 vision transformer인 Shifted window transformer이다[8]. ViT에 비해 계산량이 적고 다양한 scale(=이미지 내의 visual entity 크기)을 처리할 수 있다. Window 내 patch 수가 고정되어 있어, 이미지 크기에 선형 비례하는 계산량만 필요로 하기 때문에 여러 vision task에서 backbone net

으로 사용 가능하다. image classification 뿐만 아니라, object detection, semantic segmentation 등의 vision task의 backbone으로 사용했을 때 ViT, DeiT, RexNe(X)t보다 좋은 성능을 보였다.



Patch merging in SWIN transformer and fixed patch in ViT[8]

Swin transformer의 구조는 다음과 같다.



Swin Transformer(Tiny)의 구조 그리고 연속적인 두 transformer blocks[8]

- Patch Partitioning: 입력 rgb image를 겹치지 않는 patch로 나눈다. 각 patch는 NLP에서의 하나의 "token"과 같다.

- Stage 1: linear embedding을 먼저 거친 다음, Swin Transformer Block을 보면 MSA(multi-head self-attention) 대신 W-MSA(window MSA)와 SW-MSA(shifted window MSA)를 사용한다.

- Stage 2-4: Patch Merging 후 Transformer Block을 지나는 것을 반복한다. 맨 처음의 patch들을 점점 합쳐가면서 더 넓은 부분을 global하게 한 번에 보려는 과정. Patch

Merging을 통과하면 해상도는 2*2배 줄고 채널은 2배로 늘어난다.

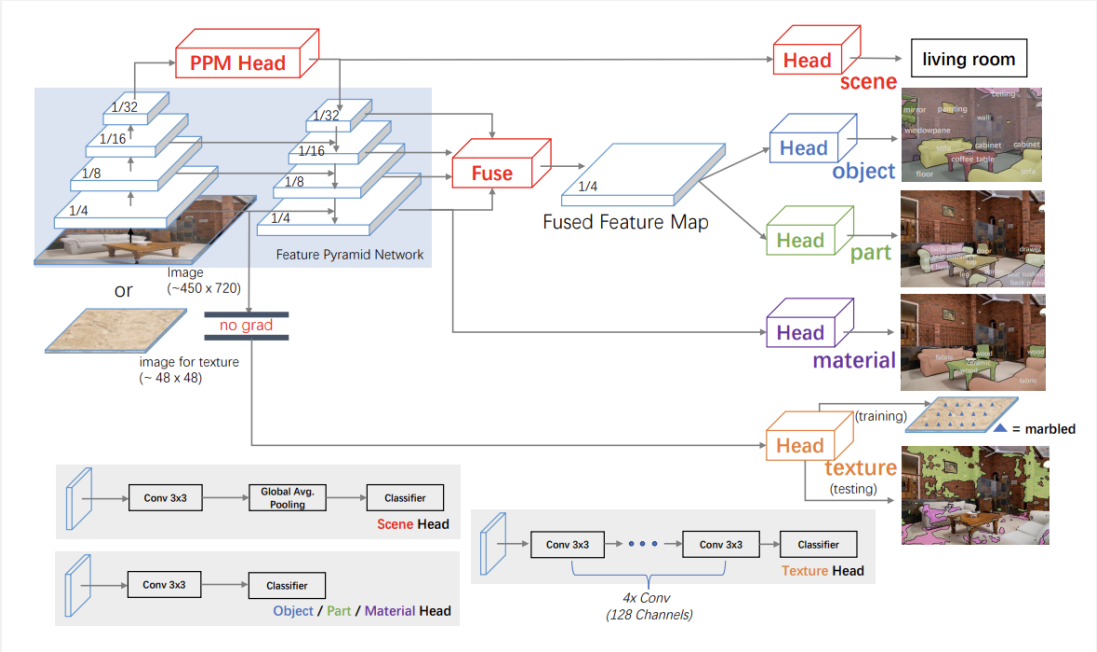
- patch merging 후 Swin transformer block을 통과하는 hierarchical 구조로 각 단계마다 representation을 갖기 때문에 다양한 scale의 entity를 다뤄야하는 image 분야에서 괜찮은 성능을 낼 수 있다.

3.5.2 UperNet

"Unified Perceptual Parsing for Scene Understanding"논문에서 소개된 모델로, 장면, 물체, 부품, 재료, 질감과 같은 객체의 세분화된 속성에 따른 segmentation을 동시에 수행한다[9].

multi-level feature를 추출하는 피라미드 구조인 FPN(Feature Pyramid Network)을 기반으로 하며, task마다 각각 다른 level의 feature를 사용한다. 마지막 layer에 PPM (Pyramid Pooling Module) Head를 연결하여 성능을 개선하고자 한다. backbone net은 공유하며, convolutional layer가 포함된 Head를 task별로 연결해 segmentation을 수행한다.

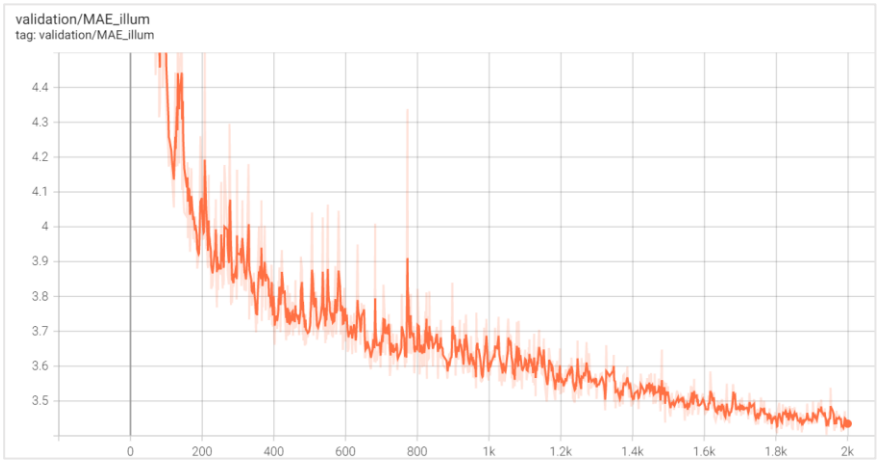
UperNet의 구조는 다음과 같다.



UPerNet framework[9]

3.5.3 LSMI dataset 학습

SWIN+UperNet[7]은 본래 semantic segmentation이 목적인 모델이기 때문에 코드의 구조를 일부 수정하였다. 원래 모델의 output인 segmentation map에는 각 픽셀마다 class일 확률이 처음에 저장되는데, regression task에 맞게 각 픽셀마다 예측된 u, v값이 저장되도록 output layer를 수정하였다. loss function은 BCE loss 대신 regression task에 자주 쓰이는 MSE loss로 바꾸었다. 모델의 arguments와 train하는 부분 또한 수정하였다.



테스트 결과는 다음과 같다.

	Mean	Median	Max
Loss	0.544799577223137	0.014143585227429867	10.94100570678711
Pred_loss	0.544799577223137	0.014143585227429867	10.94100570678711
MAE_illum	3.632069691181183	2.386161684989929	16.911975860595703
MAE_rgb	3.590916144371033	2.648771286010742	15.610945701599121
PSNR	37.941088516235354	38.269432067871094	51.60287857055664

4. 진행 시 문제점

U-net과 U-net3+ 모델 크기가 24GB 정도로 커서 GPU 3개를 사용하여 학습하였음에도 batch size를 줄여야 하는 문제가 있었다.

```
from torchsummary import summary
model = UNet_3Plus().cuda()
summary(model,(3,256,256), batch_size = 8)
```

```
==========================================================
Total params: 26,969,538
Trainable params: 26,969,538
Non-trainable params: 0
----------------------------------------------------------
Input size (MB): 6.00
Forward/backward pass size (MB): 23940.00
Params size (MB): 102.88
Estimated Total Size (MB): 24048.88
----------------------------------------------------------
```

이에 batch size를 32에서 8로 줄여 학습을 진행하였는데, 2000 epoch를 도는 데 4일 이상 소모가 되어 여러 번의 훈련을 하기 어렵다는 문제가 있었다.

5. 연구 결과

5.1 MAE

U-net, U-net3+, Swin transformer + UperNet의 성능을 MAE로 비교한 결과는 아래의 표와 같다.

모델	Min	Mean	Median	Max
U-net	0.80	**3.09**	**2.04**	15.19
U-net3+	**0.74**	3.19	2.40	**13.75**
SWIN+Upernet	1.03	3.63	2.38	16.91

다음은 각 모델들의 MAE 값을 조명의 개수 별로 분류한 결과이다.

Unet	1 illum	2 illums	3 illums
Min	**0.8073**	**0.8938**	**1.7663**
mean	3.2283	**2.9742**	**2.9133**
median	**1.8984**	**2.0474**	**2.7199**
max	12.9426	15.1927	**4.1884**

Unet3+	1 illum	2 illums	3 illums
Min	0.8136	0.9101	1.8502
mean	**3.2330**	3.2201	3.1836
median	2.3776	2.3981	3.0232
max	**9.8356**	**13.8815**	4.8143

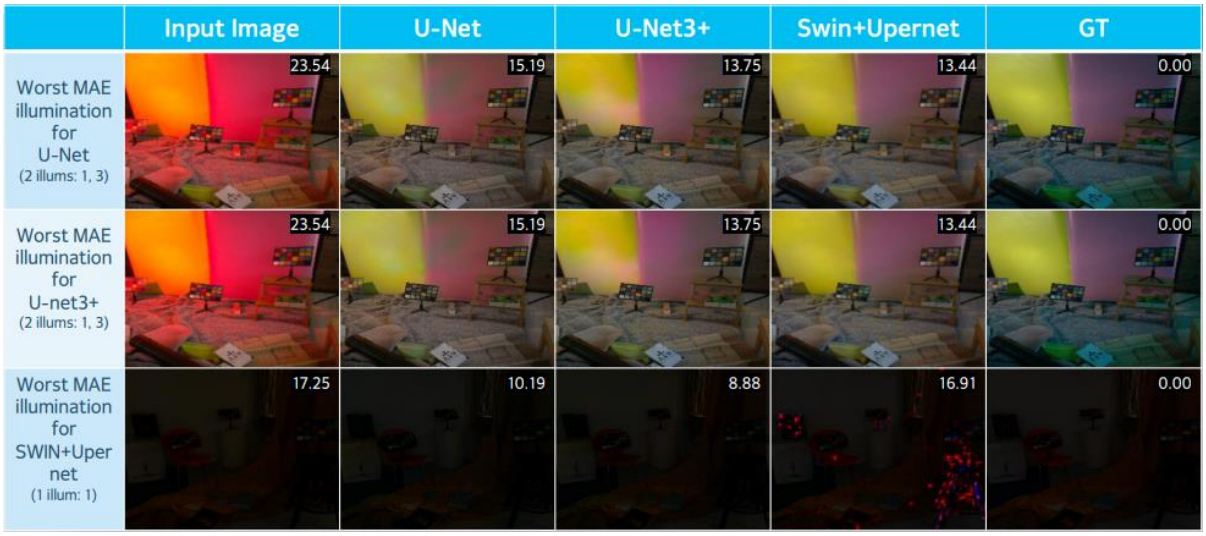
SWIN+UperNet	1 illum	2 illums	3 illums
Min	1.0359	1.0447	1.9346
mean	4.1230	3.2463	3.1117
median	2.5742	2.2858	3.139
max	16.912	16.4318	4.6149

min, mean 그리고 median 값은 baseline 모델인 U-Net이 가장 우세하였다. 그러나 max값에 있어서는 UNet3+가 가장 좋은 성능을 보였다. 이 때, 결과 이미지를 기반으로 성능 분석을 해볼 수 있다.

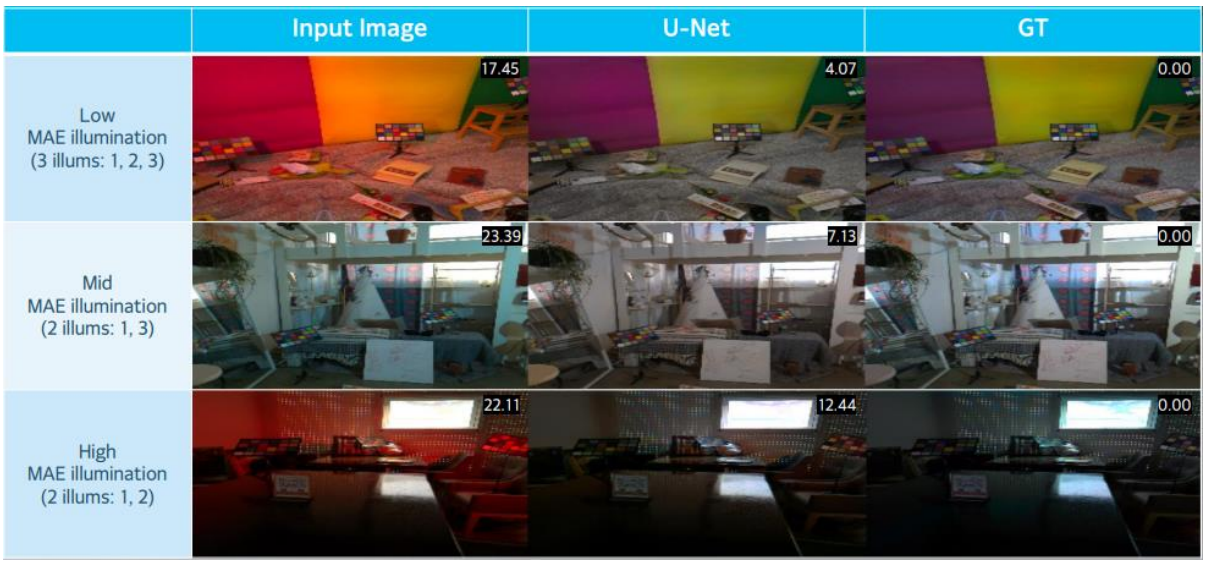
5.2 결과 이미지

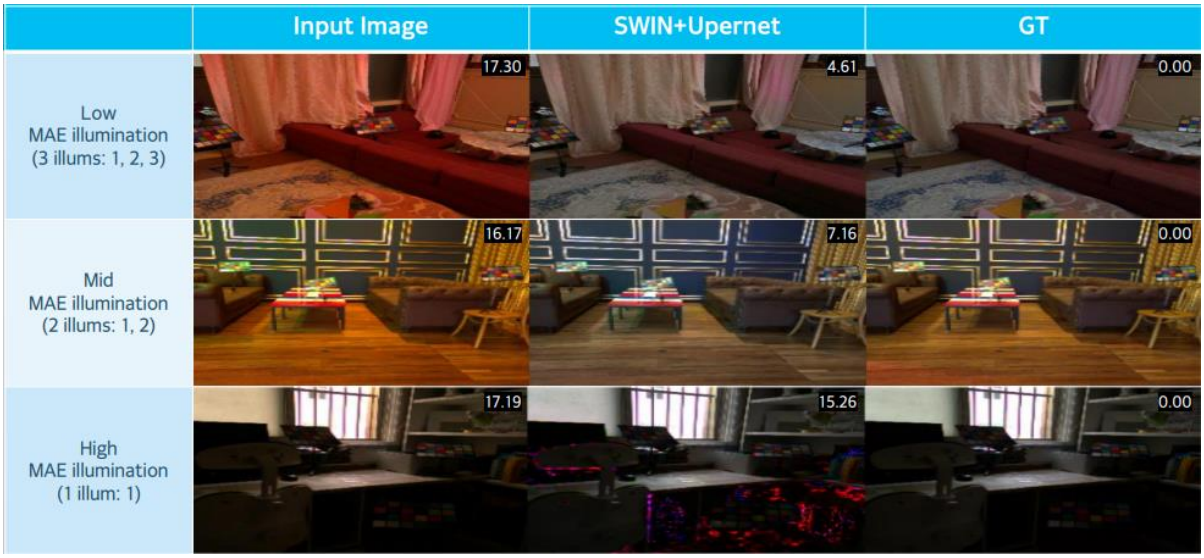
다음은 각 모델에서 가장 좋은 MAE값을 낸 이미지들의 input, output과, groundtruth이다. 각 사진의 우측 상단 숫자는 MAE값을 의미하며, 첫 번째 열의 괄호 안 숫자는 input image 내 illuminant의 개수와 그 종류를 나타낸 것이다. 1은 햇빛, 2는 천장 조명, 3은 손전등 빛을 의미한다.

다음은 각 모델에서 가장 좋지 못한 MAE값을 낸 이미지들의 input, output과, groundtruth이다.



MAE 값에 따른 output 이미지와 groundtruth 이미지의 차이를 시각적으로 확인할 수 있도록, 모델별로 MAE 값이 low, mid, high인 이미지들을 선정하였다. 다음은 U-net, U-net3+, Swin transformer + UperNet의 순서대로 이미지들의 input, output, groundtruth을 보여준다.





6. 결론 및 의의

해당 연구는 LSMI dataset을 사용하여 다중 조명 상황에서의 AWB를 위한 개선된 모델을 찾고자 하였다. 먼저 CNN 기반의 모델로 U-net3+ 모델을 사용하였고 vision Transformer인 SWIN transformer를 backbone 모델로, UperNet을 decode_head로 사용하여 pixel-level White Balance 알고리즘을 제시하였다.

AWB가 적용된 이미지를 보았을 때, 일정 수준 이상의 성능(MAE 값)을 보이는 경우에는 육안 상으로 이미지 간 큰 차이 갖지 않는다. 그러나 MAE 값이 커지는 경우, 즉 이미지 상 AWB가 적절하게 적용되지 않은 경우에는 직접 눈으로 보았을 때 이미지 간 구분 가능한 수준의 차이를 보였다. baseline 모델 U-Net과, 이번 연구를 통해 진행한 U-Net3+의 min, mean 값은 0.2도 이내의 차이를 갖는다. 이는 두 모델이 일정 수준 이상의 성능을, 육안으로 구분하기 어려운 차이로 달성했다는 것을 의미한다. 이 때 U-Net3+가 가장 낮은 max 값을 보이기 때문에, 모델의 결과물을 직접 비교

해 보았을 때에는 U-Net3+가 가장 적은 에러를 보이고 따라서 가장 준수한 결과물을 낸다고 이야기할 수 있다. 결국, 다중 조명 상황에 따른 AWB를 상용화하자는 목적 하에는 UNet3+가 가장 최적의 모델이라고 결론지을 수 있다.

차후 연구에서는 Unet3+의 skip connection 수를 줄이거나 더 큰 용량의 GPU를 사용함으로써 batch size를 키운다면 더 개선된 min, mean값을 낼 수 있을 것이다. 또한 vision regression task에 주로 사용되지 않았던 Approach임에도 불구하고 syncBN문제를 해결해 multi GPU 사용이 가능해진다면 SWIN transformer가 다중상황에서의 AWB에 대해 CNN기반의 모델과 비슷한 혹은 그 이상의 성능을 보여줄 것으로 기대해 볼 수 있을 것이라고 생각된다.

7. 역할 배분

박윤정: 연구 자료 조사, 개발, 보고서 작성, 최종 발표

유수민: 연구 자료 조사, 개발, 보고서 작성, 중간 발표

조윤수: 연구 자료 조사, 개발, 보고서 작성, 연구제안 발표

References

[1] Afifi, Mahmoud, Marcus A. Brubaker, and Michael S. Brown. "Auto White-Balance Correction for Mixed-Illuminant Scenes." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022.

[2] Kim, Dongyoung, et al. "Large Scale Multi-Illuminant (LSMI) Dataset for Developing White Balance Algorithm Under Mixed Illumination." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[3] https://github.com/DY112/LSMI-dataset

[4] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[5] Huang, Huimin, et al. "Unet 3+: A full-scale connected unet for medical image segmentation." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[6] https://github.com/ZJUGiveLab/UNet-Version

[7] https://github.com/SwinTransformer/Swin-Transformer-Semantic-Segmentation

[8] Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[9] Xiao, Tete, et al. "Unified perceptual parsing for scene understanding." Proceedings of the European Conference on Computer Vision (ECCV). 2018.