

주의 매커니즘 기반 피드백 신경망을 이용한 그림자 제거 방법

*김민우, **김원준

건국대학교

*roboticts@konkuk.ac.kr, **wonjkim@konkuk.ac.kr

Shadow Removal via Attention Mechanism and Recurrent Network

*Minwoo Kim **Wonjun Kim

Konkuk University

요약

대부분의 영상에 존재하는 그림자는 다양한 딥러닝 기반 영상처리 작업을 수행함에 방해가 되는 요소이다. 영상 내 그림자는 다양한 광원과 다양한 물체들의 상호작용에 의해 복잡하게 생성되며 이를 제거하는 것을 통해 다양한 Computer Vision task의 성능을 향상시킬 수 있다. 이 논문에서는 영상 내 그림자를 감지하여 Attention mechanism을 통해 그림자를 제거하고 Recurrent 하게 작업을 수행하며 복잡한 그림자를 단계적으로 제거하는 네트워크를 구현하였으며, Recurrent 한 네트워크에서 이전 단계의 데이터를 다음 단계에 효율적으로 전달하는 방식에 대한 실험을 수행하였다.

1. 서론

딥러닝의 발전으로 인해 사물 탐지(object detection), 추적(tracking), 인식(recognition) 등 다양한 computer vision task에 딥러닝이 활용되고 있다. 다양한 Computer vision task를 수행함에 있어 영상 내 그림자는 방해가 되는데, 영상 내 그림자 제거를 통해 Vision task의 성능을 향상할 수 있다.

그림자는 많은 영상에 존재하며, 주변 광원과 물체에 의해 생성되어 영상에 존재한다. 많은 실제 환경에서 광원은 한정적이지 않으며, 많은 물체와 복잡한 상호작용에 의해 생성된다.

이 논문에서는 그림자 감지(Shadow Detection)와 이를 통해 추출된 그림자 영역을 전달받아 그림자 제거(Shadow Removal)를 수행하는 과정을 반복하여 단계적으로 영상 내 그림자를 제거해 나가는 Attentive Recurrent[1] 네트워크와 이전 단계의 정보를 다음 단계에 효율적으로 전달하는 방법에 대하여 실험을 수행하였다.

2. 본론

2.1 Attention mechanism

그림자 제거는 전체 영상에 대해 작업을 수행하는 것이 아닌 오직 그림자 영역에 대해서만 수행한다. 그림자 제거 단계에 있어서 그림자 영역 외 영역에 대한 작업은 불필요하므로, 그림자 영역에 대한 정보를 전달하여 해당 영역에 대한 작업만을 수행하는 Attention mechanism을 사용하였다.

2.2 Recurrent 네트워크

다양한 광원과 물체로 인해 생성되는 그림자를 효과적으로 감지(detect)하고 제거(remove)하기 위해 반복적으로 두 작업을 수행하는 recurrent 한 네트워크를 설계하였다. 또한, 효과적인 작업 수행을 위해 그림자 감지기(Shadow detector)에서 이전 단계의 정보를 활용하도록 하였으며, 정보를 전달하는 두 가지 방식에 대해 실험을 수행하였다. 이 논문에서는 총 3번의 단계에 대해 실험을 수행하였다.

2.3 LSTM(Long Short Term Memory)

각 단계에서 그림자 감지기는 Convolutional layer를 통해 특징 맵(feature map)을 추출하는데, Convolutional LSTM layer를 통해 이전 단계의 특징 맵을 다음 단계로 전달하는 방식을 사용하였다. 각 단계에서는 전달받은 이전 단계의 특징 맵과 해당 단계에서 추출한 특징 맵을 합쳐 결과를 출력한다.

2.4 직접적 전달

LSTM layer를 통해 특징 맵을 다음 단계에 전달하는 것이 아닌, 이전 단계의 출력인 그림자 영역을 직접 다음 그림자 감지기의 입력에 연결하여(Concatenate) 전달하는 방식이다. LSTM을 이용한 방식과 다르게 특징 맵을 추출하기 전부터 이전 단계의 정보를 전달받는다.

3. 네트워크 설계

네트워크는 그림자 감지와 그림자 제거를 위한 두 개의 인코더로 구성되어 있으며, 두 과정을 반복하여 수행한다. 각 단계에서의 입력과 출력은 Fig.1과 같이 표현될 수 있다.

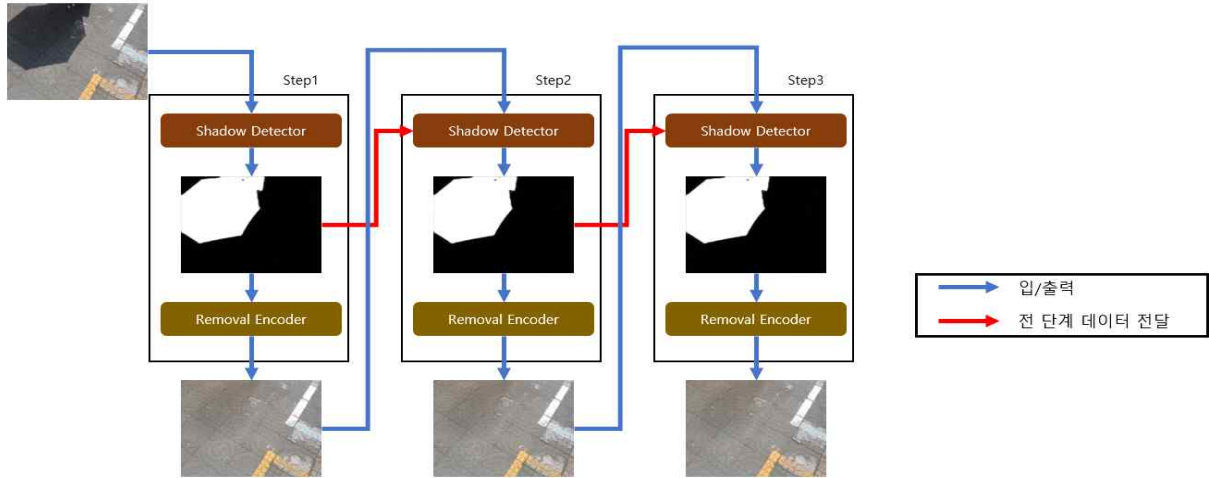


Fig. 1. 네트워크 구조.. 3번(step)의 그림자 감지 및 제거 과정을 반복한다.

3.1 그림자 감지기

그림자 제거에 있어 영상 내 제거해야 할 부분을 추출하는 것으로 그림자 제거 인코더가 제거 작업을 수행할 부분, 그림자 영역을 감지하는 역할을 수행한다. 10개의 convolutional layer로 구성되어 있으며, 최종 layer에서 sigmoid 함수를 통해 각 픽셀 마다 0에서 1 사이의 값을 출력한다. 각 픽셀의 값은 해당 픽셀이 그림자 영역에 해당하는지에 대한 score를 의미하며, 높을수록 그림자일 확률이 높은 픽셀임을 의미한다. 정보 전달 방식에 따른 감지기의 동작은 Fig.2, Fig.3과 같이 표현될 수 있다.

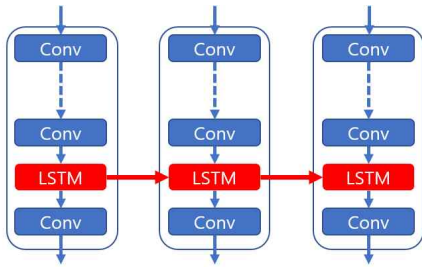


Fig.2 LSTM을 통한 그림자 감지기

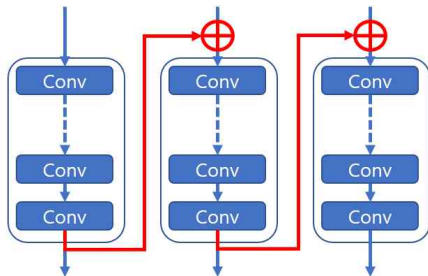


Fig.3 직접적 전달 방식의 그림자 감지기

3.2 그림자 제거 인코더

입력 영상에서 feature를 추출하고 그림자 제거 작업을 수행하는 모델이며, Convolutional layer와 Deconvolutional layer로 구성되

어 있다. 그림자 제거 인코더에서는 그림자가 제거된 영상 (Shadow-free image)를 학습하는 것이 아닌 입력 영상과의 차이인 Negative Residual[3]을 학습하여 그림자 감지 인코더의 출력인 그림자 영역에 대해 그림자 제거 작업을 수행한다.

3.3 손실함수

각 단계에서 출력된 결과에 대해 모두 손실을 계산하도록 하였으며, 각 단계에 따라 가중치를 다르게 적용하여 최종 단계의 결과에 가장 높은 가중치($\beta = 0.7$)를 설정하였다. 학습을 위해 사용한 손실 함수는 식 (2)와 같이 표현될 수 있다.

$$L_{total} = L_{det} + L_{rem} + L_{per} \quad (2)$$

Shadow detection loss(L_{det})는 각 단계에서 출력된 그림자 영역과 Ground truth의 MSE(Mean Square Error)이며, 식(3)과 같이 표현될 수 있다. A_i 는 i 번째 단계 그림자 감지기의 출력, M 은 그림자 지역의 ground truth를 의미한다.

$$L_{det} = \sum_{i=1}^3 \beta^{4-i} MSE(A_i, M) \quad (3)$$

Shadow removal loss(L_{rem})는 그림자 제거 과정에서 그림자 영역에 대한 변화만 발생하고, dataset의 Shadow image와 Shadow-free image 사이의 약간의 색상 불일치가 존재하기 때문에 그림자 영역에만 error를 계산하는 Masked-MSE를 사용하였다. Masked-MSE는 식 (4)와 같이 표현될 수 있다. O_i 는 i 번째 단계 그림자 제거기의 출력을 의미한다.

$$L_{rem} = \sum_{i=1}^3 \beta^{4-i} MSE(O_i \times M, F \times M) \quad (4)$$

Perceptual Loss(L_{per})[2]는 출력과 Ground truth 사이의 Global Difference를 VGG16 모델을 통해 계산하였으며, 식(5)와 같이 표현될 수 있다.

$$L_{per} = \sum_{i=1}^3 \beta^{4-i} MSE(VGG(O_i), VGG(F)) \quad (5)$$

4. 실험 결과

실험은 Google Colab 환경을 이용하여 수행되었으며, 네트워크 구현을 위해 pytorch를 사용하였다. 학습을 위해 ISTD dataset을 사용하였으며, 성능 평가 또한 ISTD dataset을 통해 수행하였다. 그림자 감지의 경우 BER(Balanced Error Rate)를 통해 평가를 수행하였고, 그림자 제거의 경우 LAB 색상 채널에서 RMSE(Root Mean Square Error)를 통해 평가를 수행하였다.

Table.1. 두 가지 전달 방식에 따른 실험 결과

	BER	RMSE
LSTM	7.61	5.59
직접적 전달	7.44	5.42

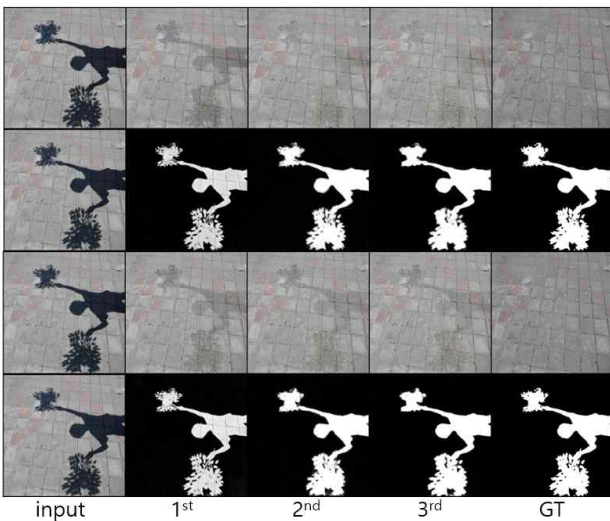


Fig.4. 단계별 출력(LSTM(상단 2줄), Concatenate(하단 2줄))

5. 결론

이 논문에서는 그림자 감지 및 제거를 위해 Attentive Recurrent 네트워크를 구현하였으며, recurrent 네트워크에서 이전 단계의 특징 맵을 전달하는 두 방식에 대한 비교를 진행하였다. 첫 번째 방식은 LSTM layer를 통해 특징 맵을 전달하는 방식으로, 각 단계에서 특징 맵을 추출한 후 이전 단계의 특징 맵과 합치는 방식이다. 두 번째 방식은 이전 단계의 출력을 다음 단계의 출력에 연결(concatenate)하는 방식으로, 이전 단계의 정보를 미리 전달받아 특징 맵을 추출하는 방식이다. Fig.5.과 같이 많은 그림자 제거가 1~2번째 단계에서 수행되는데, 2~3번째 단계에서의 그림자 감지기의 입력은 그림자 존재하지 않는 영상이 되어 출력이 현재 단계의 특징 맵보다 이전 단계의 정보에 의존하게 된다.

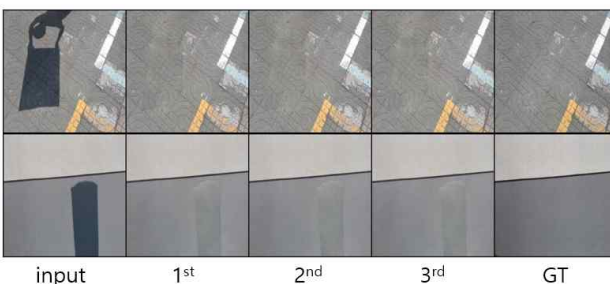


Fig.5 단계별 그림자 제거 인코더(LSTM 사용) 출력

이러한 문제점은 LSTM layer를 통해 이전 단계의 정보를 전달받는 방식 보다 입력에 직접 연결하여 전달하는 방식을 통해 의존도를 낮추어 Table.1과 같이 소폭의 성능을 향상할 수 있었다.

감사의 글

본 연구는 2021 년도 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 결과로 수행되었음 (No.2018-0-00213, SW 중심대학(건국대학교)).

참고문헌

- [1] Bin Ding, Chengjiang Long, Ling Zhang, Chunxia Xiao, "ARGAN: Attentive Recurrent Generative Adversarial Network for Shadow Detection and Removal," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10213-10222, 2019.
- [2] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European conference on computer vision. Springer, Cham, 2016.
- [3] Fu, Xueyang, et al. "Removing rain from single images via a deep detail network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.