# 컨볼루션 장기간 단기 기억 신경망의 활용과 비디오 질의응답

# An Application of Convolutional-LSTM Network and Video QA


**요 약**

기계학습 기술, 특히 딥러닝 기법의 발전은 컴퓨터 비전 분야와 자연어 처리 분야의 이해가 모두 필요한 이미지 캡셔닝 및 이미지 질의응답 분야의 발전을 이끌어냈다. 최근에는 이미지 분야를 넘어 비디오 분야에서도 컴퓨터 비전 분야와 자연어 처리 분야의 복합적 문제인 비디오 내용 질의응답 문제에 대한 연구도 이루어지고 있다. 본 논문에서는 비디오 내용에 대한 질의응답 문제를 다루는 관련 연구들을 소개하고 컨볼루션 장기간 단기 기억 신경망의 효용성을 실험을 통해 보이며 이를 사용한 질의응답 모델의 연구 방향을 제시한다.


## 1. 서 론

기계학습 기술, 특히 딥러닝 기법은 컴퓨터 비전 분야와 자연어 처리 분야의 발전을 빠른 속도로 이끌어내고 있다. 컴퓨터 비전 분야에서는 컨볼루션 인공 신경망을 활용함으로써 이미지 분류, 이미지에서의 물체 인식, 비디오에서의 동작 인식, 비디오 분류 등의 분야에서 많은 발전을 얻을 수 있었으며, 자연어 처리 분야에서는 장기간 단기 기억, GRU를 포함한 순환 인공 신경망을 활용해 문서 분류, 기계 번역 등의 분야에서 많은 발전을 이루어 냈다. 뿐만 아니라, 딥러닝 기법은 최근 컴퓨터 비전 분야와 자연어 처리 분야의 지식이 모두 필요한 이미지 캡셔닝 및 이미지 질의응답 분야에서도 고전적 모델들의 성능을 능가하며 두각을 드러내고 있다. 이는 딥러닝 기법이 기존 수작업 처리된 데이터에 의존하던 고전적인 특징 추출 기법과 달리 대용량의 데이터를 별도의 수작업 처리 없이 훈련에 사용할 수 있기 때문이다.

특히 최근 이미지 캡셔닝 및 이미지 질의응답 분야에서는 기존의 이미지 특징 데이터와 텍스트 특징 데이터를 따로 학습하던 고전적 모델에서 벗어나 이미지 특징 데이터와 텍스트 특징 데이터를 하나의 필드로 임베딩하고 이를 딥러닝 기법에 적용해 문제를 해결하는 방식이 정확도 측면에서 큰 발전을 이뤄내고 있다.

한편 딥러닝 기법은 이미지 영역과 텍스트 영역의 복합적 문제 해결 뿐 아니라 비디오 영역과 텍스트 영역의 복합적 문제 해결에도 점차 사용되고 있다. 비디오의 내용에 대한 질의응답 데이터를 제공하는 Movie QA[1]가 비디오 영역과 텍스트 영역의 복합적으로 다루는 문제의 한 예이다. 비디오에서 추출해낸 프레임 이미지만을 활용하여 해결할 수 있는 물체 인식 및 동작 인식 등의 문제와 달리, 비디오의 내용에 대한 질의응답 문제는 비디오 데이터와 함께 비디오 내용에 대한 설명을 담은 텍스트 데이터를 함께 사용하여야 한다.

본 논문에서는 비디오의 내용에 대한 질의응답 문제를 다루는 데이터셋과 관련 연구들을 소개하고, 컨볼루션 장기간 단기 기억 신경망[2]을 다른 분야에 적용했을 때의 실험 결과를 제시하고 활용한 비디오 질의응답에 대하여 다룰 것이다.

## 2. 관련 연구

### 2.1 컨볼루션 장기간 단기 기억 신경망

컨볼루션 장기간 단기 기억 신경망[2]은 본디 일기예보를 위해 사용되던 인공 신경망이다. [2]에서는 현재 기상을 나타내는 위성사진을 컨볼루션 구조를 지닌 네트워크에 입력해 특징을 찾고, 이 특징을 인코더-디코더 모델에서 사용하여 미래에 나타날 위성사진을 재구성한다. 그림 1은 도식화된 컨볼루션 장기간 단기 기억 신경망을 나타내며, 식 (1)은 컨볼루션 장기간 단기 기억 신경망의 내부 연산을 수식으로 표현한다.

$$
\begin{aligned}
i_t &= \sigma(W_\xi * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-i} + b_i) \\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-i} + b_f) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \quad (1) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
$$

식 (1)에서 $X_t$는 입력값, $C_t$는 출력되는 셀, $H_t$는 은닉 상태, $i_t$, $f_t$, $o_t$는 3D 텐서, '*'는 컨볼루션 계산, '$\circ$'는 아다마르 곱을 의미한다. 그림 1과 식(1)에서 볼 수 있듯 컨볼루션 장기간 단기 기억 신경망은 입력값과 더불어 이전의 상태와 그 출력값을 같이 입력받아 새로운 상태와 출력값을 만들어낸다.

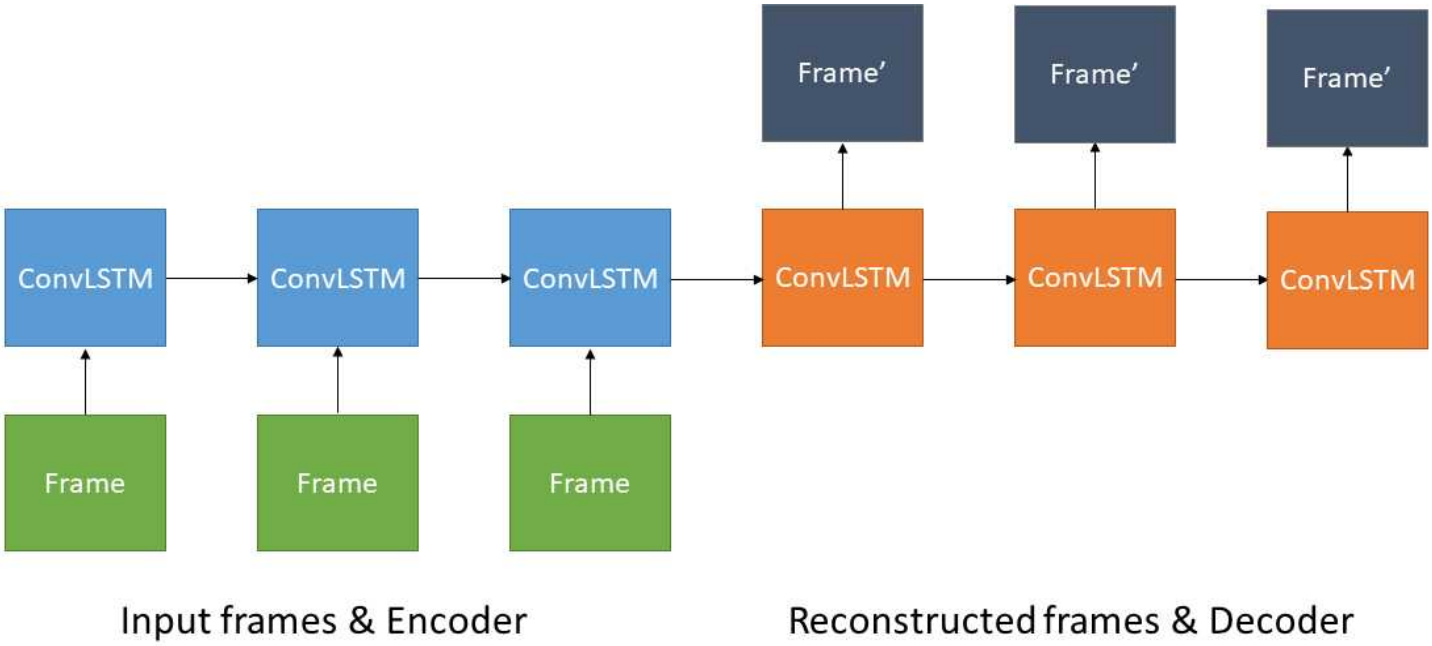


그림 1 컨볼루션 장기간 단기 기억 신경망의 도식화. 인코더 네트워크에서 입력된 프레임의 특징을 추출하고 디코더 네트워크를 사용해 프레임을 재구성하여 다시 만들어낸다.

### 2.2 비디오 내용에 대한 질의응답 문제

앞서 서론에서 언급했던 Movie QA[1]은 영화 내용에 대한 질의응답 데이터와 질의응답과 관련된 비디오 내용 데이터를 제공한다. 질의응답의 형태는 '누가 무엇을 누구에게 하였는가?' 와 같은 비교적 간단한 것부터 '누가 무엇을 어떻게 하였는가?' 나 '누가 무엇을 왜 하였는가?' 의 추상적인 질문까지 아우르는 객관식 문제이다. 이러한 질의에 답을 찾을 수 있는 데이터는 비디오, 자막, 줄거리, 화면 해설의 형태로 제공된다. 그와 동시에 제공된 비디오 내용 데이터를 활용해 답을 낼 수 있는 인공 신경망 모델 또한 제시하고 있다.

또 다른 비디오 관련 질의응답을 다루는 연구인 [3]은 설명문과 함께 제공되는 기존 비디오 데이터를 활용해 설명문의 핵심단어 위치에 빈칸을 만든 후, 빈칸에 들어갈 수 있는 단어를 객관식 보기로 제공한다. 또한 제시된 질의응답 문제에 답을 낼 수 있는 인공 신경망 모델에 대해 논하고 있다. 해당 인공 신경망 모델은 ImageNet[4]로 미리 훈련된 GoogLeNet 콘볼루션 인공 신경망[5]로 여러 장의 비디오 장면 특징들을 추출한 다음 GRU[6] 인코더-디코더 모델로 특징들 간의 맥락을 이해하여 이전 장면과 현재 장면, 미래에 나타날 장면을 재구성한다. 마지막으로 이렇게 재구성된 장면들을 Skip-thought[7]과 word2vec[8] 방식을 통해 벡터화 된 객관식 보기들의 단어와 하나의 필드로 임베딩하여 가장 가능성이 높은 보기를 답으로 선택한다.

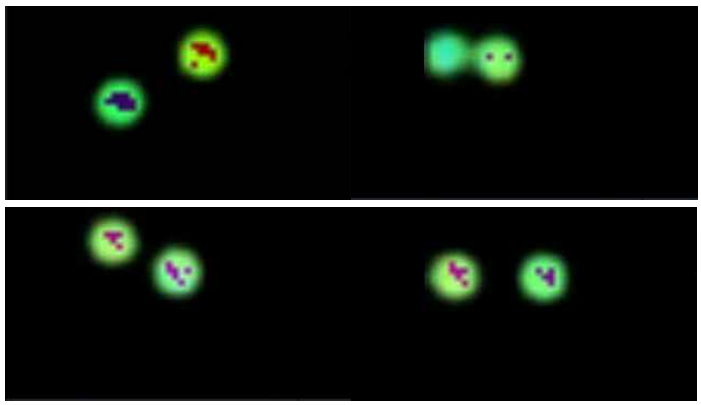

그림 2 컨볼루션 장기간 단기 기억 신경망으로 생성한 Bouncing Ball 영상의 예시. 왼쪽 위부터 시계 방향 순으로 시간의 순서를 나타내며 좌우로 움직일 때는 공이 붉은색을 띄고, 상하로 움직일 때는 공이 푸른색을 띈다.

### 3. Bouncing Ball 데이터셋 상에서의 컨볼루션 장기간 단기 기억 신경망의 활용

컨볼루션 장기간 단기 기억 인공신경망을 기상예측이 아닌 다른 분야에 적용했을 때도 좋은 결과를 얻을 수 있는지 알아보기 위해서 [9]에서 제시된 Bouncing Ball dataset을 사용하였다. 실험에 쓰인 컨볼루션 장기간 단기 기억 인공신경망은 이전에 일어났던 5장의 공 움직임 이미지를 사용해 앞으로 일어날 공의 움직임 이미지 4장을 예측하여 총 50장 분량의 공 움직임 영상을 생성한다.

실험 결과 컨볼루션 장기간 단기 기억 인공신경망은

기존의 공 이미지로부터 앞으로 일어날 공의 움직임을 잘 만들어 내는 것을 확인할 수 있었다. 그림 2의 예시에서는 공끼리 서로 부딪혔을 때 자연스럽게 반발하는 움직임을 볼 수 있었고, 공이 벽에 부딪혔을 때도 자연스러운 움직임을 볼 수 있었다. 실험에 쓰인 컨볼루션 장기간 단기 기억 인공신경망은 [10]에서 공개된 오픈소스를 기반으로 하여 구현하였다.

## 4. 컨볼루션 장기간 단기 기억 신경망을 활용한 질의응답 모델

[2]는 컨볼루션 장기간 단기 기억 신경망을 통해 이미지의 특징을 추출하고 그 결과물을 인코더-디코더 모델에서 사용하여 미래에 입력될 이미지를 재구성하는 것을 보였다. 현재 이미지를 사용해 다른 시간에 나타날 이미지를 재구성 한다는 점은 [3]에서 제시되고 있는 질의응답 문제의 해결 모델과 흡사하다. 그러나 [3]은 입력으로 받는 이미지의 특징을 더 이상 학습시키지 않고 고정되어있는 미리 훈련된 GoogLeNet 컨볼루션 인공 신경망[5]를 사용하고, [2]는 이미지에서 직접 특징을 추출하는 컨볼루션 인공 신경망을 더 좋은 결과를 얻을 수 있는 방향으로 학습시킨다는 차이점이 있다. 이러한 공통점과 차이점에서 착안하여, [2]의 컨볼루션 장기간 단기 기억 신경망의 방식을 사용해 이미지에서 직접 특징을 추출한 뒤 인코더-디코더 모델을 통해 현재 장면으로부터 이전 장면과 현재 장면, 그리고 다음 장면을 재구성할 수 있을 것으로 생각된다.

인코더-디코더 모델에 의해 재구성된 장면과 원본 장면 간의 유클리드 거리가 최소화될 수 있도록 신경망 내부의 매개 변수를 훈련시키고, 텍스트로 이루어진 질의응답을 벡터화시켜 장면 특징과 하나의 필드로 임베딩한 뒤 가장 가능성이 높은 답을 선택하는 방식으로 질의응답 문제를 해결할 수 있을 것으로 보인다.

## 5. 결론

본 논문에서는 비디오 내용에 대한 질의응답 문제를 다루는 관련 연구를 소개하고 Bouncing Ball 데이터셋 상에서 컨볼루션 장기간 단기 기억 신경망의 활용성을 실험하여 그 결과를 통해 효용성을 확인했다. 또한 컨볼루션 장기간 단기 기억 신경망을 사용한 질의응답 모델의 연구 방향을 제시하였다.

## 6. 참고문헌


[1] Tapaswi, Makarand, et al. “Movieqa: Understanding stories in movies through question-answering.“ Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] Xingjian, S. H. I., et al. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting.“ Advances in neural information processing systems. 2015.

[3] Zhu, Linchao, et al. “Uncovering the temporal context for video question answering.“ International Journal of Computer Vision 124.3 (2017): 409-421.

[4] Russakovsky, Olga, et al. “Imagenet large scale visual recognition challenge.“ International Journal of Computer Vision 115.3 (2015): 211-252.

[5] Szegedy, Christian, et al. “Going deeper with convolutions.“ Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2015.

[6] Cho, Kyunghyun, et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation.“ arXiv preprint arXiv:1406.1078 (2014).

[7] Kiros, Ryan, et al. “Skip-thought vectors.“ Advances in neural information processing systems. 2015.

[8] Goldberg, Yoav, and Omer Levy. “word2vec explained: Deriving mikolov et al.’s negative-sampling word-embedding method.“ arXiv preprint arXiv:1402.3722 (2014).

[9] Sutskever, Ilya, Geoffrey E. Hinton, and Graham W. Taylor. “The recurrent temporal restricted boltzmann machine.“ Advances in Neural Information Processing Systems. 2009.

[10] https://github.com/loliverhennigh/Convolutional-LSTM-in-Tensorflow