AdapTex: 하이브리드 비전 트랜스포머 모델을 사용한 Image-to-LaTeX 수식 OCR 모델과 어댑터를 적용한 전이학습

AdapTex: Image-to-LaTeX Equation OCR Model with Hybrid Vision Transformer and Transfer Learning with Adapter

요 약

광학 문자 인식은 이미지 속 문자를 기계가 인식할 수 있는 형태로 변환하는 작업이다. 한편 학술 및 교육 분야에서 수식을 전자문서에서 표현하기 위해 LaTeX 형식으로 변환하는 과정은 번거로운 작업이다. 본 연구는 이러한 과정을 딥러닝 모델을 통해 자동화하여 연구자, 교육자 및 학생들의 업무 효율성을 높이고자 한다. 이를 위해 수식 이미지를 입력 받아 LaTeX 형식의 시퀀스를 출력하는 트랜스포머 기반의 모델을 제시하고 데이터셋을 구축하여 학습하였다. 해당 모델은 기존 모델에 비해 수식 이미지의 인식 성능이 향상되었으며, 어댑터 추가를 통한 효율적 학습과 성능 개선을 검증하였다.

1. 서 론

광학 문자 인식(Optical Character Recognition)은 이미지 속 문자를 기계가 인식할 수 있는 형태로 변환하는 작업이며, 차량 번호판 인식에서부터 모바일 촬영을 통한 서류 제출까지 다양한 프로세스의 효율화 및 자동화에 사용된다. 한편 학술 및 교육 분야에서 수식은 논문, 교재 및 보고서의 핵심적인 요소이나, 이를 전자문서에서 표현하기 위해 LaTeX 형식으로 변환하는 과정은 번거로운 작업이다. 본 연구는 이러한 과정을 딥러닝 모델을 통해 자동화하여 연구자, 교육자 및 학생들의 업무 효율성을 높이고자 한다.

딥러닝 기반의 문자 인식 시스템은 CNN과 RNN을 기반으로 하는 CRNN 모델[1]이 있으며, 해당 모델은 CNN을 통해 이미지에서 특성 시퀀스를 추출하고, 양방향 LSTM을 통해 문자열을 예측한다. 이후 정확도와 효율성을 위해 어텐션 기반의 인코더와 디코더를 사용하는 모델이 제시되었으며[2], 어텐션에 적합한 이미지 임베딩 기법을 사용하기도 한다[3], 비전 분야에서 ViT[4]가 소개된 이후에는 문자 인식 분야에서도 ViT 인코더를 도입한 모델이 제시되며 성능이 개선되었다[5].

한편 수식을 인식하고 LaTeX 형식으로 변환하는 과제에 관한 연구는 Deng et al.[6]에 의해 제시되었다. 해당 연구에서는 어텐션을 포함한 RNN 기반의 디코더를 사용하였고, 나아가 ViT 인코더를 도입한 모델은 오픈소스 프로젝트[7]를 통해 제시되었다. 그러나 해당 연구와 프로젝트에서는 인쇄체에 해당하는 수식 이미지, 즉 LaTeX 형식을 통해 렌더링된 수식 이미지에 대해서만 인식하는 한계가 존재했다.

본 연구에서는 인쇄체뿐만 아니라 손글씨로 작성된 수식 이미지도 인식할 수 있는 모델을 개발하고자 한다. 이를 위해 하이브리드 ViT를 인코더로 채택하고, 어텐션을 포함한 RNN 기반의 디코더로 구성된 모델을 사용하였다. 본 연구에서 사용된 모델은 오픈소스 프로젝트[7]의 소스코드를 기반으로 구현하였고, 사전 학습된 모델을 사용하였다. 사전 학습 모델의 성능을 개선하기 위해 데이터셋을 추가하여 파인튜닝을 진행하였고, 비교 연구(Ablation study)를 통해 최적 파라미터를 탐색하였다. 또한 효율적인 성능 개선을 위하여 인코더에 어댑터(AdaptFormer)[8]를 추가하고 전이 학습을 진행하였다.

수식 인식 과제에서 본 연구가 기여한 바는 다음과 같다. 첫째, 하이브리드 ViT 인코더를 사용한 모델의 손글씨 수식 이미지의 인식 성능을 검증하였고, 인쇄체와 손글씨 수식 이미지의 인식 성능을 개선시켰다. 둘째, 문자 인식 분야에서 어댑터 추가를 통한 효율성 및 성능 개선을 검증하였다. 마지막으로 수식 인식 모델의 소스코드 공개 및 데모 프로그램 배포를 통해 추후 OCR 분야에서 수식 인식 시스템의 고도화에 기여하고자 하며, LaTeX 형식으로 수식 변환이 필요한 연구자, 교육자 및 학생들의 업무 효율성에 도움이 되고자 한다.

2. 수식 OCR 모델

2.1 인코더

본 연구에서는 2,3,7층의 레이어 블록으로 구성된 ResNetV2 백본과 ViT를 사용한 하이브리드 구조의 인코더를 사용하였다. 이미지를 입력받으면, 사이즈를 16으로 하는 패치 임베딩 후 백본 네트워크를 통과한다. 출력된 특징 맵은 포지셔널 임베딩 후, 멀티-헤드 어텐션과 MLP로 구성된 트랜스포머 인코더를 통과한다[그림 1(a)].

패치 사이즈가 16이기에 인풋 이미지의 최소는 32x32, 최대는 672x192로 설정하고 32배수로 리사이징하였다. 인코더의 차원은 256, 깊이는 4와 6, 헤드 수는 8로 설정하였다.

2.2 디코더

디코더는 기존 트랜스포머 디코더와 동일한 디코더를 사용하였고, LaTeX 토큰의 시퀀스를 출력한다. 토크나이저는 구축한 데이터셋의 LaTeX 정답 토큰을 이용하여 생성하였다. 패딩은 사용하지 않았으며, 최대 시퀀스 길이는 512, 차원은 256, 깊이는 4, 헤드수는 8로 설정하였다.

2.3 어댑터(AdaptFormer)

본 연구에서 사용된 어댑터인 AdaptFormer[8]는 사전 학습된 가중치는 동결하고 새로 추가된 어댑터 부분만 학습하는 기법이다. 매우 적은 파라미터를 사용하여 풀-파인튜닝과 동일하거나 그 이상의 성능을 보여준다.

어댑터는 인코더에서 멀티-헤드 어텐션을 통과한 피쳐를 다운사이징 후 ReLU를 거치고 업사이징하는 구조로 구성되어있다. 이후 MLP의 출력과 잔차 연결, 어댑터의 출력를 합쳐 최종 출력이 얻어진다[그림 1(b)].

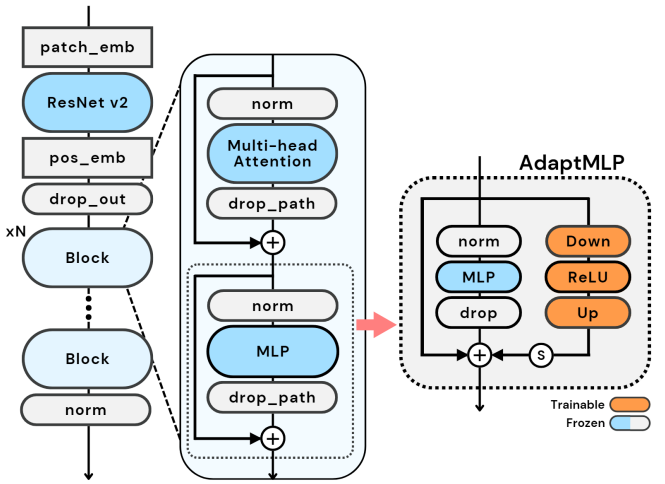


그림 1: 수식 인식 모델 구조

3. 데이터

본 연구에서는 수식 이미지와 LaTeX 형식으로 라벨링된 데이터셋을 수집하였고, 인쇄체와 손글씨 형식의 데이터 균형을 맞추어 통합된 데이터셋을 구축하였다. 통합 과정에서 특정 수식에 해당하는 LaTeX 표현의 정답(Ground Truth)을 통일하기 위한 LaTeX 토큰의 정제를 진행하였다. 데이터셋의 명명은 표 1을 따른다.

표 1: 실험 데이터의 명명 및 정보

Dataset	명명	형태	크기
PDF	P1	인쇄체	234,884(235k)
AIHUB_pdf	P2	인쇄체	24,559(25k)
CROHME	H1	손글씨	10,846(11k)
AIHUB_handwritten	H2	손글씨	88,605(88k)
AIDA	H3	손글씨	100,000(100k)
CROHME_symbol	S	손글씨	375,974(376k)

PDF 데이터셋은 KDD cup[9]에서 구축된, 논문의 수식을 LaTeX 형식으로 라벨링하고 렌더링을 통해 인쇄체 형식의 이미지를 생성한 데이터셋이다. CROHME는 ICDAR와 ICFHR[10]에서 구축된 손글씨 데이터셋이다. AIHUB는 국내 데이터셋 플랫폼[11]에서 제공하는 초, 중, 고등학교의 OCR 데이터셋이며, 라벨링된 바운딩박스를 통해 수식 이미지를 추출하였다. AIDA는 Kaggle[12]에서 제공하며, 미적분 수식 데이터셋이다. CROHME_symbol에 대해서는 후술한다.

4. 연구 방법

4.1 데이터셋 파인튜닝

파인튜닝은 인쇄체 형식인 PDF 데이터셋으로 사전 학습된 모델을 풀-파인튜닝 하는 방식을 채택하였다. 인쇄체 이미지와 손글씨 이미지의 형태가 상이하여, 모델의 모든 가중치를 학습시키는 풀-파인튜닝 방식이 효과적임을 사전 실험을 통해 검증하였다. 일반화 성능을 위해 데이터셋을 순차적으로 추가하며 인쇄체와 손글씨 이미지에 대한 인식 성능 추이를 확인하였다.

또한, 수식에서 ‘s’와 ‘5’와 같이 유사한 문자(symbol)를 구분하지 못하는 문제를 개선하기 위해, CROHME 데이터셋에서 단일 토큰에 해당하는 문자를 추출한 CROHME_symbol 데이터셋을 생성하여 학습한 후 성능을 비교하였다.

4.2 최적 파라미터 탐색

학습률(learning rate)은 모델의 성능에 큰 영향을 주며, 파인튜닝의 경우 일반적으로 낮은 수치를 설정하는 것이 효과적이다. 최적의 학습률을 적용하기 위해 학습률 스케줄러의 최대-최소 값, 그래디언트 클리핑의 임계값을 변경하여 비교 실험(Ablation study)를 진행하였다. 또한 사전 학습된 모델의 특성을 보존하기 위해 인코더-디코더 학습률의 비율을 변경하여 학습 후 성능을 비교하였다.

4.3 네트워크 구조 변경(AdapTex 모델)

모델의 구조적 한계로 인한 언더피팅 현상을 개선하기 위해 어텐션 블록의 깊이를 4에서 6으로 변경하고, AdaptFormer[8]를 적용하여 효율성 및 성능 검증을 진행하였다. 해당 실험의 경우 파인튜닝된 모델 중 가장 좋은 성능을 보이는 모델을 새롭게 베이스 모델로 채택하여 진행하였다.

5. 실험 및 결과

모델은 PyTorch로 구현되었고 크로스 엔트로피를 손실로 하며 AdamW 옵티마이저를 사용하였다. GPU는 Tesla T4를 사용하였다.

5.1 평가

평가 지표는 기계번역 분야에서 주로 사용되는 지표를 세가지 선정하였다. (1) N-gram 유사도 기반의 BLEU 스코어, (2) 추론된 LaTeX 토큰 정확도, (3) 문자열이 같아지기 위한 최소 편집거리(Edit distance)가 이에 해당한다. 각 테스트 데이터셋에 대한 평가 지표를 데이터셋 크기별로 가중 평균하여 모델의 성능을 비교하였다.

5.2 데이터셋 파인튜닝

P1 데이터셋으로 사전 학습된 모델[7]에서 데이터셋이 순차적으로 추가되면서 성능이 향상되었고, 최종적으로는 인쇄체와 손글씨 데이터에 대한 성능이 비슷한 수치를 보인다. 반면에 심볼 데이터셋 학습의 경우 예상과는 달리 성능이 저하됨을 확인하였다[표 2].

표 2: 데이터셋 파인튜닝 평가 지표

	인쇄체 수식 이미지			손글씨 수식 이미지		
Dataset	BLEU	Token acc	Edit dist↓	BLEU	Token acc	Edit dist↓
P1*	0.878	0.586	0.092	-	-	-
(+)H1	0.889	0.596	0.086	0.498	0.457	2.910
(+)H2	0.877	0.579	0.097	0.786	0.674	0.228
(+)P2	**0.917**	**0.736**	0.056	0.784	0.685	0.191
(+)H3**	0.916	0.732	**0.053**	**0.912**	**0.867**	**0.077**
(+)S	0.895	0.685	0.069	0.752	0.612	0.241

* P1 데이터셋으로 사전 학습된 모델[7], ** 인코더 깊이 6

5.3 최적 파라미터 탐색

기본 설정은 StepLR 스케줄러를 사용하였으며, 시작 학습률 1.0E-3, 그래디언트 클리핑의 임계값은 1.0으로 설정하였다(Model 1). 학습률의 경우 StepLR 보다 CAWR(Cosine Annealing Warm Restarts) 스케줄러를

사용하는 것이 좋은 성능을 보이며, 최소-최대 범위는 1.0E-7에서 1.0E-3(Model 2), 그래디언트 클리핑의 임계값은 0.5로 설정하는 것(Model 4)이 효과적임을 확인했다(Model 3의 경우 CAWR 스케줄러를 1.0E-6에서 1.0E-4로 설정). 반면에 인코더-디코더 학습률의 비율(10:1)을 조절하는 경우(Model 5)는 예상과는 달리 성능이 개선되지 않았다[표 3].

표 3: 파라미터 비교 실험 평가 지표

	인쇄체 수식 이미지			손글씨 수식 이미지		
Model	BLEU	Token acc	Edit dist↓	BLEU	Token acc	Edit dist↓
1	0.917	0.736	0.056	0.784	0.685	0.191
2	0.914	0.729	0.057	**0.806**	**0.704**	0.195
3	0.887	0.666	0.075	0.674	0.523	0.387
4	**0.916**	**0.740**	**0.056**	0.805	0.700	**0.180**
5	0.906	0.701	0.062	0.672	0.547	0.342

5.4 네트워크 구조 변경(AdapTex 모델)

이전 실험에서 파인튜닝한 베이스모델(Model 1)에 AdaptFormer를 추가하여 전이 학습을 진행한 결과 언더피팅 현상이 개선됨을 확인하였다(Model 2). 어텐션 블록의 깊이를 4에서 6으로 변경한 경우에도 손글씨 데이터의 인식 성능이 소폭 향상되었다(Model 3). 또한 심볼 데이터셋의 학습에서 성능이 저하되었던 문제(Model 4)도 일부 개선되었다(Model 5). 이를 통해 가중치 동결 후 추가된 어댑터의 가중치를 학습하는 것만으로도 성능 개선이 가능함을 검증하였다[표4].

표 4: AdaptFormer 적용 모델 평가 지표

	인쇄체 수식 이미지			손글씨 수식 이미지		
Model	BLEU	Token acc	Edit dist↓	BLEU	Token acc	Edit dist↓
1	0.917	0.736	0.056	0.784	0.685	0.191
2	**0.921**	**0.745**	0.055	0.901	0.840	0.085
3	0.917	0.739	**0.054**	**0.933**	**0.887**	**0.059**
4	0.895	0.685	0.069	0.752	0.612	0.241
5	0.914	0.735	0.057	0.929	0.880	0.067

6. 결론

본 연구에서는 공개된 사전 학습 모델[7]의 수식 인식 성능을 개선하였으며, 기존 모델의 한계점인 손글씨 수식에 대한 인식도 가능하도록 모델을 개발하였다. 이를 위해 하이브리드 ViT 인코더와 AdaptFormer를 적용한 트랜스포머 기반의 네트워크를 제시하였고, 최적 파라미터를 탐색하였으며, 어댑터의 추가를 통한 효율적인 학습으로 성능 개선이 가능함을 검증하였다.

본 연구의 목표는 해당 모델을 통해 수식 이미지를 LaTeX 형식으로 변환하여 연구자, 교육자 및 학생들의 업무 효율성을 높이는 것이다. 실험 결과를 바탕으로 데모 모델을 개발하여 오픈소스화하고 배포를 진행할 예정이며, 데모 모델의 테스트 샘플은 [부록]에 첨부하였다.

참고 문헌

[1] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[2] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," *CoRR*, vol. abs/1709.02054, 2017.

[3] J. Lee, S. Park, J. Baek, S. J. Oh, S. Kim, and H. Lee, "On recognizing texts of arbitrary shapes with 2d self-attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 546–547, 2020.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] P. Lyu, C. Zhang, S. Liu, M. Qiao, Y. Xu, L. Wu, K. Yao, J. Han, E. Ding, and J. Wang, "Maskocr: text recognition with masked encoder-decoder pretraining," *arXiv preprint arXiv:2206.00311*, 2022.

[6] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *International Conference on Machine Learning*, pp. 980–989, PMLR, 2017.

[7] L. Blecher. (2022). [online], LaTeX-OCR:https://github.com/ lukasblecher/LaTeX-OCR. (Accessed: 2023.Aug).

[8] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16664–16678, 2022.

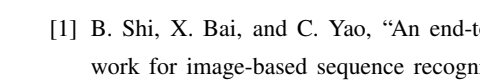
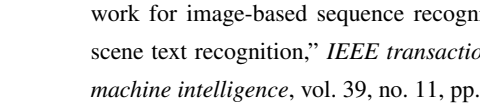
[9] A. Kanervisto. (2016). [Dataset], im2latex-100k: arXiv:1609.04938. (downloaded: 2023.Aug).

[10] H. Mouchère. (2011-2013). [Dataset], CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions, ICDAR, ICFHR. (downloaded: 2023.Aug).

[11] (주)씨유박스. (2021). [Dataset], 수식, 도형 낙서기호 OCR 데이터: https://aihub.or.kr. (downloaded: 2023.Aug).

[12] Pearson. (2020). [Dataset], Aida Calculus Math Handwriting Recognition Dataset: https://www.kaggle.com/aidapearson/ocr-data. (downloaded: 2023.Sep).

부 록

이미지 input	모델 output(LaTeX 수식 렌더링 시)
$\Psi \approx \psi_1(q)\vert(x' + \frac{1}{2}p_1'^2 t, k')) + \psi_2(q)\vert(x' + \frac{1}{2}p_2'^2 t, k')).$	$\Psi \approx \psi_1(q)\vert(x' + \frac{1}{2}p_1'^2 t, k')) + \psi_2(q)\vert(x' + \frac{1}{2}p_2'^2 t, k')).$
$x^n = \sum_{k=1}^{n} \left\{ n \atop k \right\} x^{\underline{k}} = \sum_{k=1}^{n} \left\{ n \atop k \right\} (-1)^{n-k} x^{\overline{k}},$	$x^n = \sum_{k=1}^{n} \left\{ n \atop k \right\} x^{\underline{k}} = \sum_{k=1}^{n} \left\{ n \atop k \right\} (-1)^{n-k} x^{\overline{k}},$
	$\frac{d}{dt} e^{aX(t)} = \int_0^1 e^{(1-\alpha)X(t)} \frac{dX(t)}{dt} e^{\alpha X(t)} d\alpha$
	$\int \frac{\sin(x)+1}{\sqrt{\cos^3(x)+\tan(x)}} dx$