Scaling Up Image-to-LaTeX Performance: Sumen An End-to-End Transformer Model With Large Dataset

Trung Hoang Quoc[1], Bao Thai Duy[1], Trung Nguyen Quoc[1], Huu-Thanh Duong[2], and Vinh Truong Hoang[2]

[1] Department of Information Technology Specialization, FPT University, Ho Chi Minh City, Vietnam,
{trunghqse151140,baotdse161680,trungnq46}@fpt.edu.vn
[2] Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam
{thanh.dh,vinh.th}@ou.edu.vn

Abstract. Recognizing mathematical formulas in images and translating them into LaTeX sequences, both printed and handwritten, is challenging due to the complexity of two-dimensional formulas and lack of training data. Traditional methods only handle simple formulas and are not very effective for complex formulas. In this paper, we introduce the Sumen (**S**caling **U**p I**m**age-to-LaT**e**X Performa**n**ce) model, an encoder-decoder architecture-based model in Transformer with attention mechanism trained on the largest dataset from previous works. The model achieves a BLEU score of 95.59, Edit Distance (ED) of 97.3, and Exact Match (EM) of 69.23 on the img2latex100k benchmark, and corresponding Expression Recognition Rates (ExpRate) of 58.01/82.39/78.99 on CROHME 2014/2016/2019. All of our metrics outperform previous state-of-the-art methods on both printed and handwritten formulas.

Keywords: Image to LaTeX, printed formula recognition, handwritten formula recognition, CROHME, Img2latex-100k

1 Introduction

Mathematical formulas are essential for life, appearing daily and applied in many fields such as science, technology, articles, and websites. Typing and representing mathematical formulas with markup languages is relatively difficult and error-prone. Converting math formula images to LaTeX, also known as img2latex, was created to help scientists, teachers, and non-professionals easily convert math formulas from images to LaTeX sequences and represent them in their documents. Methods like the INFTY system [25] have solved simple mathematical formulas such as superscripts, subscripts, special symbols, and fractions, but they still cannot solve complex formulas because they require the model to not only recognize characters but also understand the relationship between characters like fractions.

In recent years, encoder-decoder architecture-based methods have been widely used and achieved good results. The encoder-decoder simplifies the end-to-end model, takes the image input for processing, understands and captures important information in the image, and then represents this information as a feature vector, which is then fed into the decoder and directly generates the output without the need to complicate the data like segmenting the text locations in the image. Although this architecture has been working well in recent years, previous methods have only focused on Printed Mathematical Expression Recognition (PMER) or Handwritten Mathematical Expression Recognition (HMER), but they have not been very good at both tasks in the same method. We have continued to develop, improve, and supplement our efforts to create a model that can support users in converting images to LaTeX sequences for both printed and handwritten formulas. The primary contributions in this paper are:

- Introducing a large-scale dataset for mathematical formula recognition.
- Proposing the Sumen model, which achieves state-of-the-art results compared to existing methods on both PMER and HMER tasks.
- Releasing the source code and checkpoints of the Sumen model for the research community to use and further develop.

2 Related Works

The advent of deep neural networks has replaced classical methods with encoder-decoder architectures and brought about good results and real success in the field of Computer Vision & Natural Language Processing. Methods using encoder-decoder architectures such as [11] propose a neural encoder-decoder with a coarse-to-fine attention mechanism. The authors use a Convolutional Neural Network (CNN) encoder to extract features from the image and an Recurrent Neural Network (RNN) decoder implements a conditional language model over the vocabulary. [23] introduces a neural transducer model with visual attention, which uses a CNN as an encoder and an RNN as a decoder, combined with beam search during inference. [29] proposes a method that includes a CNN combined with positional encoding used in the encoder to extract features. The features are augmented with 2D positional encoding before being unfolded into a vector and fed into Long Short Term Memory (LSTM) decoder to translate into a sequence of LaTeX tokens. [37] proposes a model that applies a Transformer-based encoder-decoder architecture. The encoder uses a Vision Transformer (ViT) and takes inspiration from machine translation to apply to the img2latex task. Additionally, this method combines the use of a YOLO model [22] for the preprocessing step of separating single-line formulas from multi-line formulas to improve the model's accuracy. BTTR [36] and ABM [7] methods introduce a novel bidirectional training strategy with the aim of learning LaTeX sequences from left-to-right and right-to-left directions on the RNN decoder to solve the lack of coverage problem [34]. However, this leads to more parameters and longer training time. Inspired by the coverage mechanism in RNN, CoMER [35] proposes a model that improves the Transformer's shortcomings regarding the lack of coverage problem.

It uses an Attention Refinement Module (ARM) to refine attention weights with past alignment information without hurting its parallelism and performs better than the vanilla transformer decoder and RNN decoder in the HMER task.

3 Methods

The architecture we use in this paper is Nougat [8], which is a transformer-based encoder-decoder architecture. Nougat is a Visual Transformer model that performs an Optical Character Recognition (OCR) task to understand scientific documents and translate them into markup language. The encoder is Swin Transformer [17] and the decoder is decoder-Transformer [27]. The entire architecture of the model is shown in Fig.1. We initialize the weights from Nougat.

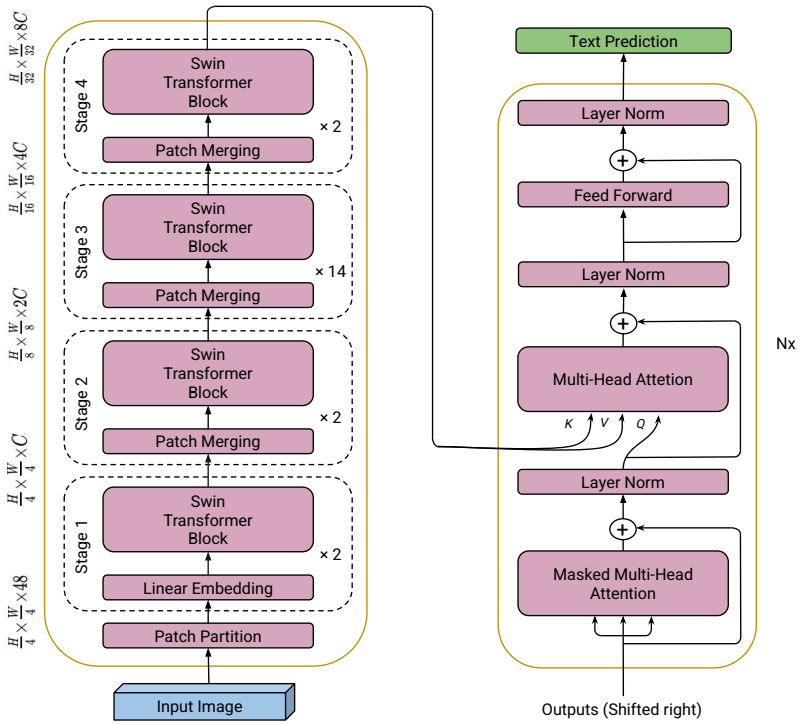


Fig. 1: Overview of sumen architecture (left: encoder, right: decoder)

Encoder: The advent of Swin Transformer is a breakthrough in the field of computer vision, combining the power of the attention mechanism and the hierarchical architecture of CNN. In which width and height are reduced and channels are increased in the later layers, providing flexibility to scale different image sizes. Some notable features are shown in Fig.2 as follows: Window-based

self-attention is used to attend to patches or tokens in a window (local attention). However, sharing information across different windows is crucial to understanding the relationships between objects in the image. Therefore, shifted windows allow windows to communicate, which is an idea based on stride in CNN but with a different variation called cyclic shift. The model takes an input of an image with size $H \times W \times 3$, which is then used to extract important information from the image and output the encoder's KV feature vectors.

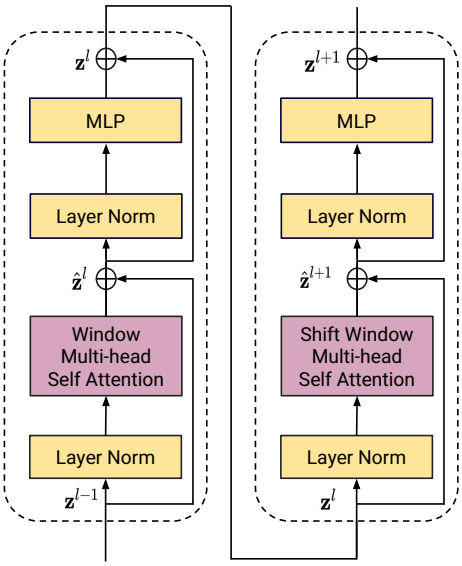


Fig. 2: Swin transformer block

Decoder: The decoder uses decoder-Transformer, which is an auto-regressive language modeling (unidirectional), meaning that the model reads words only in the left-to-right direction. The decoder receives important information in the image represented by feature vectors KV from the encoder through cross multi-head attention and generates LaTeX sequences corresponding to the formulas in the image.

4 Experiment

4.1 Dataset

In the formula recognition task, we divide it into two domains: Handwritten and printed mathematical formulas. We use different datasets to train and evaluate

the model based on these two domains. In it, we collect and build the largest dataset to date from online sources, creating a robust and well-generalizable dataset. This dataset consists of approximately 3.4 million image-text pairs, including both handwritten mathematical expressions (200,330 samples) and printed mathematical expressions (3,237,250 samples).

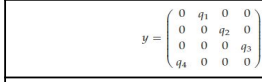
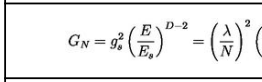
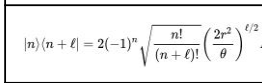
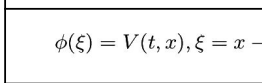
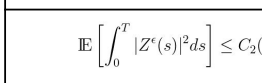
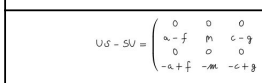
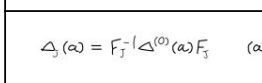
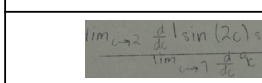
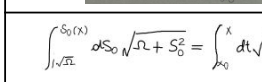
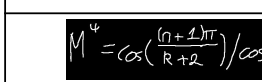
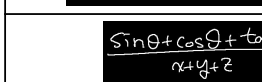
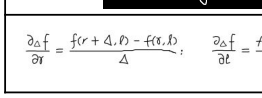
Images	Labels
	y = \left(\begin{array}{ c c c c }{{0}}&{{q_{1}}}&{{0}}&{{0}} \\ {{0}}&{{0}}&{{q_{2}}}&{{0}} \\ {{0}}&{{0}}&{{0}}&{{q_{3}}} \\ {{q_{4}}}&{{0}}&{{0}}&{{0}} \end{array} \right)
	G_{N} = g_{s}^{2} \left(\frac{E}{E_{s}} \right)^{D-2} = \left(\frac{\lambda}{N} \right)^{2} \left(\frac{E}{E_{s}} \right)^{D-2-2\alpha}
	\mid n \rangle \langle n + \ell \mid = 2 (- 1) ^ { n } \sqrt { \frac { n ! } { (n + \ell) ! } } \left(\frac { 2 r ^ { 2 } } { \theta } \right) ^ { \ell / 2 } L _ { n } ^ { \ell } (2 r ^ { 2 } / \theta) e ^ { - r ^ { 2 } / \theta } e ^ { i \ell \varphi } \, ,
	\begin{array} { r } { \phi (\xi) = V (t , x) , \xi = x - c t , c > 0 } \end{array}
	\displaystyle { \mathrm { I \! E } } \left[\int _ { 0 } ^ { T }	Z ^ { \epsilon } (s)	^ { 2 } d s \right] \leq C _ { 2 } (1 + | x | ^ { 2 }) ,
	U S - S U = \left(\begin{array} { c c c c } { 0 } & { 0 } & { 0 } & { 0 } \\ { a - f } & { m } & { c - g } & { m } \\ { 0 } & { 0 } & { 0 } & { 0 } \\ { - a + f } & { - m } & { - c + g } & { - m } \end{array} \right) .
	\Delta _ { \mathrm { J } } (\alpha) = F _ { \mathrm { J } } ^ { - 1 } \Delta ^ { (0) } (\alpha) F _ { \mathrm { J } } \quad (\alpha \in U (\mathfrak { s l } (2))) .
	\frac { \operatorname* { l i m } _ { c \to 2 } \frac { d } { d c } 7 \sin \left(2 c \right) } { \operatorname* { l i m } _ { c \to 7 } \frac { d } { d c } 9 c }
	\int _ { i \sqrt { \Omega } } ^ { \delta _ { 0 } (x) } d S _ { 0 } \sqrt { \Omega + S _ { 0 } ^ { 2 } } = \int _ { x _ { 0 } } ^ { x } d t \sqrt { \Lambda + \varphi ^ { 2 } (x) }
	M ^ { 4 } = \cos (\frac { (n + 1) \pi } { R + 2 }) / \cos (\frac { \pi } { R + e })
	\frac { \sin \theta + \cos \theta + \tan \theta } { x + y + z }
	\frac { \partial _ { \Delta } f } { \partial r } = \frac { f (r + \Delta , \ell) - f (r , \ell) } { \Delta } ; \qquad \frac { \partial _ { \Delta } f } { \partial \ell } = \frac { f (r , \ell + \Delta) - f (r , \ell) } { \Delta } \, .

Fig. 3: Data examples

Printed mathematical expressions: We collect from Im2latex-100k dataset [10], I2L-140K Normalized dataset and Im2latex-90k Normalized dataset [23], Im2latex-170k dataset [1], Im2latex-230k dataset [2], latex-formulas dataset [3] and Im2latex dataset [4].

Handwritten mathematical expressions: We collected data from the Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) dataset [19, 20, 18], Aida Calculus Math Handwriting Recognition Dataset [5] and Handwritten Mathematical Expression Convert LaTeX [6]

Pre-processing: Due to the large dataset and the fact that the same mathematical formula can be represented in different LaTeX string formats in an image, it is easy to cause polymorphic ambiguity. To address this issue, we use the normalization method with KaTeX parser [13]. We convert the raw LaTeX strings into an abstract syntax tree, and then apply safe normalizing tree transformation to eliminate ambiguity in the LaTeX markup strings. Some previous works using this method include [10, 29].

Model	BLEU	Edit Distance	Exact Match
INFTY [25]	66.65	96.4	59.6
CNNENC [10]	75.01	61.17	53.53
WYGIWYS [10]	87.73	87.60	77.46
MI2LS w/o Reinforce [29]	89.08	91.09	79.39
MI2LS with Reinforce [29]	90.28	92.28	82.33
DoubleAttention [29]	88.42	88.57	79.81
DenseNet [28]	88.25	91.57	-
MathBERT [21]	90.45	90.11	**87.52**
Zhou *et al.* [37]	92.11	90.0	60.2
Wang *et al.* [28]	85.71	90.25	28.68
DenseNet(2 blocks)	85.82	91.38	35.68
C-S attention	86.54	90.75	31.79
DenseNet + C-S	88.25	91.57	37.09
Sumen-base (ours)	**95.59**	**97.3**	69.23

Table 1: Comparison of performance printed formula recognition task with previous methods on Im2latex-100k test set.

4.2 Metrics

We use the following metrics for testing to evaluate and compare model results with previous methods: Bilingual Evaluation Understudy (BLEU), Edit Distance (ED), Exact Match (EM) and Expression Recognition Rates (ExpRate).

4.3 Implementation Details

The entire model is implemented using Pytorch & Transformer framework. During training, we use the Lion optimizer [9] with weight decay = 1e-2, $\beta_1 = 0.95$ and $\beta_2 = 0.98$. We use a maximum length of 512, image resolution of 224×768 (height $\times$ width), and a learning rate of 1e-4 with 1k warmup steps based on a cosine schedule. The entire model is trained for 10 epochs with a batch size of 2048 (gradient accumulation) on 1 GPU Tesla P100-PCIE-16GB.

Dataset	Model	ExpRate	$\leq$ 1 error	$\leq$ 2 error	$\leq$ 3 error
CROHME 14	DenseWAP [32]	43.0	57.8	61.9	-
	DenseWAP-TD [33]	49.1	64.2	67.8	-
	WS-WAP [26]	53.65	-	-	-
	Li *et al.* [16]	56.59	69.07	75.25	78.60
	Ding *et al.* [12]	58.72	-	-	-
	BTTR [36]	53.96	66.02	70.28	-
	BTTR (CoMER) [35]	55.17	67.85	72.11	74.14
	CoMER [35]	**59.33**	71.70	75.66	77.89
	PAL [31]	39.66	56.80	68.51	-
	WAP [34]	46.55	61.16	65.21	66.13
	PGS [15]	48.78	66.13	73.94	-
	PGS-v2 [30]	48.88	64.50	69.78	-
	DLA [14]	49.85	-	-	-
	ABM [7]	56.85	**73.73**	**81.24**	-
	WYGIWYS [10]	36.4	-	-	-
	Sumen-base (ours)	58.01	72.11	80.22	**85.04**
CROHME 16	DenseWAP [32]	40.1	54.3	57.8	-
	DenseWAP-TD [33]	48.5	62.3	65.3	-
	WS-WAP [26]	51.96	64.34	70.10	72.97
	Li *et al.* [16]	54.58	69.31	73.76	76.02
	Ding *et al.* [12]	57.72	70.01	76.37	78.90
	BTTR [36]	52.31	63.90	68.61	-
	BTTR (CoMER) [35]	56.58	68.88	74.19	76.90
	CoMER [35]	59.81	74.37	80.30	82.56
	WAP [34]	44.55	57.10	61.55	62.34
	PGS [15]	36.27	-	-	-
	PGS-v2 [30]	49.61	64.08	70.27	-
	DLA [14]	47.34	-	-	-
	ABM [7]	52.92	69.66	78.73	-
	Sumen-base (ours)	**82.39**	**89.97**	**94.42**	**95.99**
CROHME 19	DenseWAP [32]	41.7	55.5	59.3	-
	DenseWAP-TD [33]	51.4	66.1	69.1	-
	Ding *et al.* [12]	61.38	75.15	80.23	82.65
	BTTR [36]	52.96	65.97	69.14	-
	BTTR (CoMER) [35]	59.55	72.23	76.06	78.40
	CoMER [35]	62.97	77.40	81.40	83.07
	ABM [7]	53.96	71.06	78.65	-
	Sumen-base (ours)	**78.99**	**86.22**	**90.5**	**92.07**

Table 2: Comparison of performance handwritten formula recognition task with previous methods on CROHME 2014/2016/2019 test sets based on expression recognition rates score.

4.4 Comparison with Prior Works

We use the Im2latex-100k dataset for our experiments on PMER task, which is a collection of about 100,000 real-world mathematical expressions rendered from public papers on the arxiv.org server. This is a popular dataset dedicated to the PMER task, and the test set consists of 10,285 samples. We use metrics such as BLEU score (4-gram), EM score, and ED score to evaluate on this dataset. The results shown in the Table 1 indicate that our model performs very well overall in terms of the entire string sequence on BLEU and ED, with the highest result, while EM is only at an average level and not the highest.

In addition, we use the CROHME dataset to demonstrate the effectiveness of our model on the HMER task, which is a large open dataset for handwritten mathematical expressions. The test set consists of three versions: CROHME 2014 (986 samples), CROHME 2016 (1,147 samples), CROHME 2019 (1,199 samples). We choose expression level metrics: ExpRate (%), ExpRate ≤ 1 error (%), ExpRate ≤ 2 error (%), and ExpRate ≤ 3 error (%) provided by the CROHME 2019 organizers [18]. The results shown in the Table Table 2 and indicate that we achieve the best results on 4/4 ExpRate metrics on CROHME 16&19. However, on CROHME 14 we only achieve the best result on 1/4 ExpRate metrics. Overall, our model outperforms other models.

5 Conclusion

We have proposed a transformer-based encoder-decoder architecture for converting images containing mathematical formulas into LaTeX sequences. We also introduce a new large dataset that helps scale up the accuracy of img2latex performance. Compared with other models, we have successfully addressed the major challenge of recognizing mathematical formulas, including both printed and handwritten formulas in the same model. Our model achieves state-of-the-art performance on the CROHME 2016/2019 and Im2latex-100k test sets. Future work will concentrate on creating a large handwritten mathematical expressions dataset to balance the current dataset using GAN as proposed in [24] to convert printed mathematical expression images into handwritten mathematical expression images.

References

[1] https://www.kaggle.com/datasets/rvente/im2latex170k.
[2] https://www.kaggle.com/datasets/gregoryeritsyan/im2latex-230k.
[3] https://huggingface.co/datasets/OleehyO/latex-formulas.
[4] https://huggingface.co/datasets/AlFrauch/im2latex.
[5] https://www.v7labs.com/open-datasets/aida.
[6] https://huggingface.co/datasets/Azu/Handwritten-Mathematical-Expression-Convert-LaTeX.

[7] Xiaohang Bian et al. “Handwritten Mathematical Expression Recognition via Attention Aggregation based Bi-directional Mutual Learning”. In: *CoRR* abs/2112.03603 (2021). arXiv: 2112.03603. URL: https://arxiv.org/abs/2112.03603.

[8] Lukas Blecher et al. “Nougat: Neural Optical Understanding for Academic Documents”. In: *CoRR* abs/2308.13418 (2023). DOI: 10.48550/ARXIV.2308.13418. arXiv: 2308.13418. URL: https://doi.org/10.48550/arXiv.2308.13418.

[9] Xiangning Chen et al. “Symbolic Discovery of Optimization Algorithms”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh et al. 2023. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/9a39b4925e35cf447ccba8757137d84f-Abstract-Conference.html.

[10] Yuntian Deng, Anssi Kanervisto, and Alexander M. Rush. “What You Get Is What You See: A Visual Markup Decompiler”. In: *CoRR* abs/1609.04938 (2016). arXiv: 1609.04938. URL: http://arxiv.org/abs/1609.04938.

[11] Yuntian Deng et al. “Image-to-Markup Generation with Coarse-to-Fine Attention”. In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 980–989. URL: http://proceedings.mlr.press/v70/deng17a.html.

[12] Haisong Ding, Kai Chen, and Qiang Huo. “An Encoder-Decoder Approach to Handwritten Mathematical Expression Recognition with Multi-head Attention and Stacked Decoder”. In: *Document Analysis and Recognition – ICDAR 2021*. Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. Cham: Springer International Publishing, 2021, pp. 602–616. ISBN: 978-3-030-86331-9.

[13] *Katex*. https://katex.org/.

[14] Anh Duc Le. “Recognizing handwritten mathematical expressions via paired dual loss attention network and printed mathematical expressions”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2413–2418. DOI: 10.1109/CVPRW50498.2020.00291.

[15] Anh Duc Le, Bipin Indurkhya, and Masaki Nakagawa. “Pattern generation strategies for improving recognition of Handwritten Mathematical Expressions”. In: *Pattern Recognit. Lett.* 128 (2019), pp. 255–262. DOI: 10.1016/J.PATREC.2019.09.002. URL: https://doi.org/10.1016/j.patrec.2019.09.002.

[16] Zhe Li et al. “Improving Attention-Based Handwritten Mathematical Expression Recognition with Scale Augmentation and Drop Attention”. In: *17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020, Dortmund, Germany, September 8-10, 2020*. IEEE, 2020, pp. 175–180. DOI: 10.1109/ICFHR2020.2020.00041. URL: https://doi.org/10.1109/ICFHR2020.2020.00041.

[17] Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *CoRR* abs/2103.14030 (2021). arXiv: 2103.14030. URL: https://arxiv.org/abs/2103.14030.

[18] Mahshad Mahdavi et al. "ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection". In: *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*. IEEE, 2019, pp. 1533–1538. DOI: 10.1109/ICDAR.2019.00247. URL: https://doi.org/10.1109/ICDAR.2019.00247.

[19] Harold Mouchère et al. "ICFHR 2014 Competition on Recognition of On-Line Handwritten Mathematical Expressions (CROHME 2014)". In: *14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014*. IEEE Computer Society, 2014, pp. 791–796. DOI: 10.1109/ICFHR.2014.138. URL: https://doi.org/10.1109/ICFHR.2014.138.

[20] Harold Mouchère et al. "ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions". In: *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016*. IEEE Computer Society, 2016, pp. 607–612. DOI: 10.1109/ICFHR.2016.0116. URL: https://doi.org/10.1109/ICFHR.2016.0116.

[21] Shuai Peng et al. "MathBERT: A Pre-Trained Model for Mathematical Formula Understanding". In: *CoRR* abs/2105.00377 (2021). arXiv: 2105.00377. URL: https://arxiv.org/abs/2105.00377.

[22] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[23] Sumeet S. Singh. "Teaching Machines to Code: Neural Markup Generation with Visual Attention". In: *CoRR* abs/1802.05415 (2018). arXiv: 1802.05415. URL: http://arxiv.org/abs/1802.05415.

[24] Matthias Springstein, Eric Müller-Budack, and Ralph Ewerth. "Unsupervised Training Data Generation of Handwritten Formulas using Generative Adversarial Networks with Self-Attention". In: *CoRR* abs/2106.09432 (2021). arXiv: 2106.09432. URL: https://arxiv.org/abs/2106.09432.

[25] Masakazu Suzuki et al. "INFTY: an integrated OCR system for mathematical documents". In: *Proceedings of the 2003 ACM Symposium on Document Engineering, Grenoble, France, November 20-22, 2003*. ACM, 2003, pp. 95–104. DOI: 10.1145/958220.958239. URL: https://doi.org/10.1145/958220.958239.

[26] Thanh-Nghia Truong et al. "Improvement of End-to-End Offline Handwritten Mathematical Expression Recognition by Weakly Supervised Learning". In: *17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020, Dortmund, Germany, September 8-10, 2020*. IEEE, 2020, pp. 181–186. DOI: 10.1109/ICFHR2020.2020.00042. URL: https://doi.org/10.1109/ICFHR2020.2020.00042.

[27] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[28] Jian Wang, Yunchuan Sun, and Shenling Wang. "Image To Latex with DenseNet Encoder and Joint Attention". In: *Procedia Computer Science* 147 (2019). 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, pp. 374–380. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.01.246. URL: https://www.sciencedirect.com/science/article/pii/S1877050919302686.

[29] Zelun Wang and Jyh-Charn Liu. "Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training". In: *Int. J. Document Anal. Recognit.* 24.1 (2021), pp. 63–75. DOI: 10.1007/S10032-020-00360-2. URL: https://doi.org/10.1007/s10032-020-00360-2.

[30] Jin-Wen Wu et al. "Handwritten Mathematical Expression Recognition via Paired Adversarial Learning". In: *Int. J. Comput. Vis.* 128.10 (2020), pp. 2386–2401. DOI: 10.1007/S11263-020-01291-5. URL: https://doi.org/10.1007/s11263-020-01291-5.

[31] Jin-Wen Wu et al. "Image-to-Markup Generation via Paired Adversarial Learning". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Michele Berlingerio et al. Cham: Springer International Publishing, 2019, pp. 18–34. ISBN: 978-3-030-10925-7.

[32] Jianshu Zhang, Jun Du, and Lirong Dai. "Multi-Scale Attention with Dense Encoder for Handwritten Mathematical Expression Recognition". In: *CoRR* abs/1801.03530 (2018). arXiv: 1801.03530. URL: http://arxiv.org/abs/1801.03530.

[33] Jianshu Zhang et al. "A Tree-Structured Decoder for Image-to-Markup Generation". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 11076–11085. URL: http://proceedings.mlr.press/v119/zhang20g.html.

[34] Jianshu Zhang et al. "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition". In: *Pattern Recognit.* 71 (2017), pp. 196–206. DOI: 10.1016/J.PATCOG.2017.06.017. URL: https://doi.org/10.1016/j.patcog.2017.06.017.

[35] Wenqi Zhao and Liangcai Gao. "CoMER: Modeling Coverage for Transformer-Based Handwritten Mathematical Expression Recognition". In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 392–408. ISBN: 978-3-031-19815-1.

[36] Wenqi Zhao et al. "Handwritten Mathematical Expression Recognition with Bidirectionally Trained Transformer". In: *CoRR* abs/2105.02412 (2021). arXiv: 2105.02412. URL: https://arxiv.org/abs/2105.02412.

[37] Mingle Zhou et al. "An End-to-End Formula Recognition Method Integrated Attention Mechanism". In: *Mathematics* 11.1 (2023). ISSN: 2227-7390. DOI: 10.3390/math11010177. URL: https://www.mdpi.com/2227-7390/11/1/177.