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], Tien Nguyen Quoc[1], and Vinh Truong Hoang[2]

[1] FPT University, Ho Chi Minh City, Vietnam,
{trunghqse151140,baotdse161680,trungnq46,tiennq27}@fpt.edu.vn
[2] Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam
vinh.th@ou.edu.vn

Abstract. The recognition of mathematical formulas, both printed formula recognition and handwritten formula recognition, is a challenging task due to the complexity of two-dimensional formulas and the lack of data for training. Traditional methods previously only handled simple formulas and were not very effective for complex formulas. Recent methods have mainly used encoder-decoder architectures, such as CNN-RNN, CNN-LSTM, and ViT-Transformer. In this paper, we introduce Sumen (**S**caling **U**p I**m**age-to-LaT**e**X Performa**n**ce), a model based on the encoder-decoder architecture in Transformer with self-attention mechanism. It is trained on the largest dataset from previous works, which we synthesized from various sources. The model achieves a BLEU score of 94.51, Edit Distance (ED) of 96.59, and Exact Match (EM) of 61.81 on the img2latex100k benchmark. For CROHME 2014, 2016, and 2019, the corresponding Word Error Rates (WER) are 9.07, 3.9, and 5.88, and the Expression Recognition Rates (ExpRate) are 57.71, 76.02, and 72.97. All of our metrics have been shown to outperform previous state-of-the-art methods on both printed and handwritten formulas. The source code and data are available at `https://github.com/hoang-quoc-trung/sumen-latex-ocr`.

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

1 Introduction

In recent years, Encoder-Decoder architectures have achieved great success in the field of Computer Vision & Natural Language Processing intersection, known as multi-modal, such as Image Captioning, Speech to Text, and Text to Image. In these tasks, the encoder is responsible for the vision module and the decoder is responsible for the language module. The image is processed, understood, and important information is captured, then represented as a dense feature vector. These feature vectors are then fed into the decoder to generate the desired string sequences. Commonly used architectures for the encoder include CNN, Vision

Transformer (ViT) [11], and for the decoder include RNN, LSTM, Transformer [35].

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 using markup languages is relatively difficult and error-prone. Convert math formula image to LaTeX (img2latex), also known as image-to-text, was born to help scientists, teachers, and non-professionals convert mathematical formulas from images to Latex code and represent them in their documents more easily.

Formula recognition task is a very challenging task. Methods like INFTY system [32] have solved simple mathematical formulas such as superscript, subscript, special symbols, and fractions, but they have not yet solved complex formulas because they require the model to understand the relationship between characters, such as fractions, not just recognize characters. The advent of deep neural networks has brought good results, with encoder-decoder simplifying the end-to-end model, taking image input and directly generating output without the need to complicate the data like segment location of text segments in the image. Although it has been a well-performing architecture in recent years, previous methods have focused only on printed formula recognition task [8, 9, 31, 36, 37, 47] or handwritten formula recognition [45, 4, 46, 16, 6, 41, 42]. However, they have not been effective on both of these tasks in the same method. Based on the achievements of encoder-decoder architecture, we have continued to develop, improve, and supplement to create a model that supports users in converting images to Latex for both printed and handwritten formulas.

Decoder-based RNN has the disadvantages of vanishing gradient and exploding gradient, challenging with long sequences, slow computation, and only learning short-term memory [28]. For complex mathematical formula markup sequences that can be up to hundreds of LaTeX tokens, the hidden state vector in RNN is not enough to compress all the information from the encoder. LSTM was born to solve the problems arising from RNN to be able to scaled up capture long-term memory when needed, mitigating the vanishing gradient problem and can forget unimportant information through the forget gate. However, LSTM is complex, requires large time and computational resources, and cannot be parallelized. Transformer was born based on the self-attention mechanism and gradually replaced methods like RNN, LSTM due to the convenience of parallel processing and overcoming the above limitations. It is widely used with Large Language Model (LLM) and achieves state-of-the-art results.

Recognizing the potential of Transformer to achieve good results in many fields, we decided to use Transformer for both encoder and decoder. We use Swin Transformer [22] as the vision module and decoder transformer for the language module.

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 printed and handwritten formulas.

– Releasing the source code and checkpoints of the Sumen model for the research community to use and further develop.

2 Related Works

Zanibbi *et al.* [40] proposed a method using Baseline Structure Tree (BST) with 3 steps: layout pass used to identify the position and order of symbols, lexical pass used to group symbols into units such as decimals, function names and operators, and finally expression analysis pass used to convert the expression into a usable string format. Lee *et al.* [18] used the method of segmenting and understanding text, mathematical expressions in documents, including 6 stages: page segmentation and labeling, character segmentation, feature extraction, character recognition, expression formation, and error correction and expression extraction. Berman *et al.* [3] proposed a method to improve the accuracy of 3-phase methods (segmentation, remove obvious noise, primary recognition) using the principle of image-connected.

Wang *et al.* [37] proposed a novel model following the encoder-decoder architecture including CNN in the encoding used in the encoder to extract features, the features will be augmented with 2D positional encoding before being unfolded into a vector and LSTM with the soft attention mechanism used in the decoder to translate the encoder output into a sequence of LaTeX tokens. With 2 training steps: step 1 using the Maximum-Likelihood Estimation (MLE) to train each token individually, step 2 train on the entire LaTeX sequence with the policy gradient algorithm from reinforcement learning. The model overcomes the exposure bias problem by closing the feedback loop in the decoder during sequence-level training.

Bian *et al.* [4] proposed a novel model instead of using attention-based encoder-decoder models as nowadays, the new model is Attention aggregation based Bi-directional Mutual learning Network (ABM) which includes: one shared encoder and two parallel inverse decoders (L2R and R2L) using mutual distillation to improve the two decoders in utilizing complementary information from two directions, Attention Aggregation Module (AAM) to deal with mathematical symbols with different scales, and finally only using the L2R branch for inference, to keep the speed and model size.

Zhou *et al.* [47] proposed a model based on the encoder-decoder architecture, with the encoder being a Vision Transformer (ViT) and the decoder using an attention-based Transformer. The model uses joint codec training and Cross-Entropy as a loss function, which achieves relatively stable results.

Zhao *et al.* [45] proposes a model that improves the Transformer's deficiency in suffering from the lack of coverage problem. The model adopts the coverage information in the transformer decoder and uses the Attention Refinement Module (ARM) to refine attention weights with past alignment information without affecting parallelism. It also adds self-coverage and cross-coverage and utilizes the past alignment information from the current and previous layers.

3 Methods

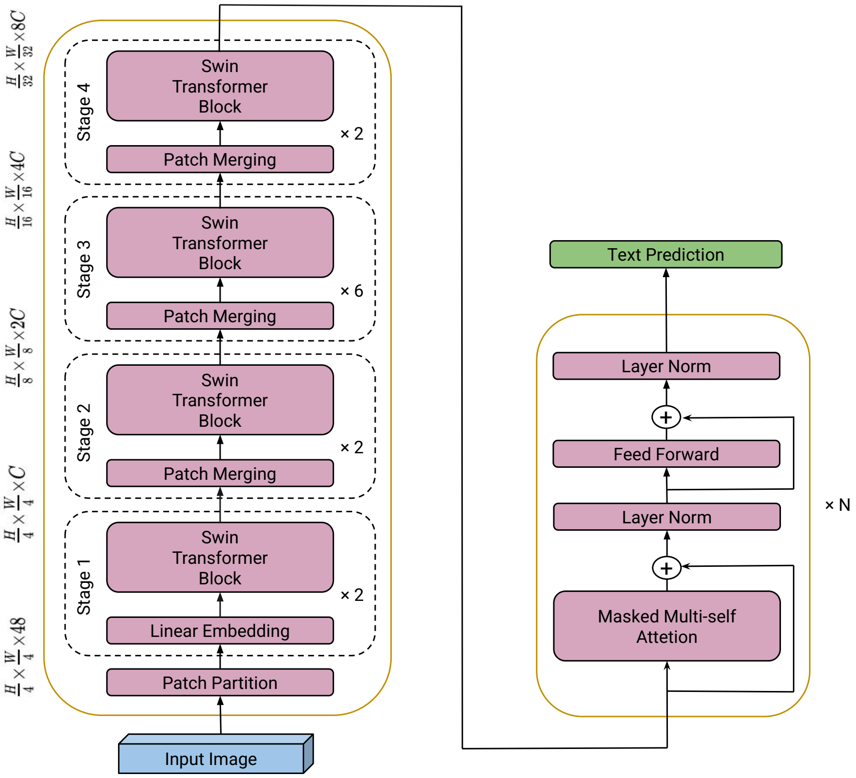


Fig. 1: Overview of sumen architecture (left: encoder, right: decoder)

The architecture we use in this paper is the end-to-end encoder-decoder transformer architecture [35], specifically the Nougat transformer model [5] proposed by Meta AI, which inherits and develops from the Donut architecture [14]. 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 and the decoder is mBART-decoder [21]. The whole architecture of the model is shown in Fig.1. We initialize the weights from Nougat.

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 $H \times W \times 3$, then it will be used to extract important information from the image and output a sequence of the encoder features z.

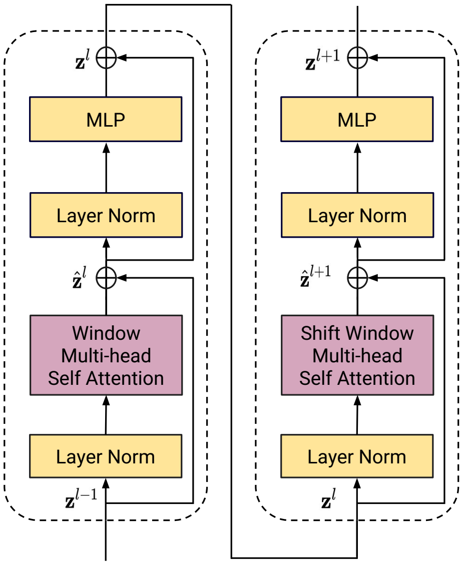


Fig. 2: Swin transformer block

Decoder: The decoder uses decoder-mBART, which is an auto-regressive transformer (unidirectional), meaning that the model reads words only in the left-to-right direction. The decoder takes the representation produced by the encoded image z and reconstructs LaTeX respectively using masked multi-self attention to align between the vector z and target logits to generate the output probability of the best tokens. Then, to convert these generated tokens into text, they need to be mapped to the vocabulary v through an index.

Data Augmentation: We employ the same image augmentation methods as Nougat [5] during training, including RGB shift, bitmap, shift scale rotate, grid distortion, affine, elastic transform, random brightness contrast, image compression, gauss noise, gaussian blur.

4 Experiment

4.1 Dataset

In the formula recognition task, we divide it into two domains: printed formula recognition and handwritten formula recognition. They use different test sets to evaluate the model. We collect and construct the largest dataset to date from online sources, creating a dataset that is robust and generalizable. This dataset consists of approximately 3.4 million image-text pairs, including both handwritten mathematical expressions (200330 samples) and printed mathematical expressions (3237250 samples). The percentage of data is illustrated in Fig.3 and data details are shown in Fig.4.

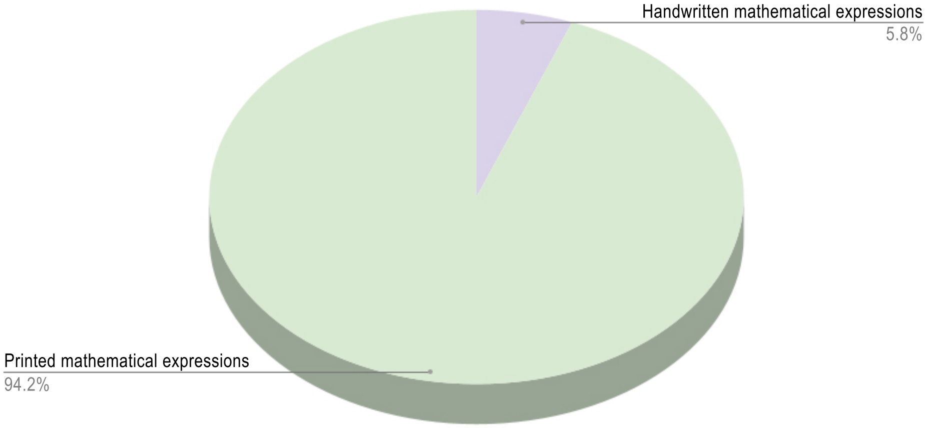


Fig. 3: Overall dataset

Printed mathematical expressions: We collect from Im2latex-100k dataset [8], I2L-140K Normalized dataset and Im2latex-90k Normalized dataset [31], Im2latex-170k dataset [30], Im2latex-230k dataset [12], latex-formulas dataset [26] and Im2latex dataset [1].

Handwritten mathematical expressions: We collected data from the Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) dataset [24, 25, 23], Aida Calculus Math Handwriting Recognition Dataset [34] and Handwritten Mathematical Expression Convert LaTeX [2]

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 [8, 37].

4.2 Metric

The following metrics are used for the test dataset:

- **Bilingual Evaluation Understudy (BLEU):** Compares the similarity between the machine translation and the reference translation. BLEU uses n-grams (combinations of n words) to measure the level of overlap between text strings [27].
- **Edit Distance (ED)**: Or Levenshtein distance [19], measures the minimum number of edit operations required to transform one text string into another. Edit operations include insertion, deletion, and substitution of characters. The result is calculated by dividing the total number of operations performed by the total number of words in the reference. To compare with other methods, we take $1-$ minimum ED.
- **Exact Match (EM)**: EM checks whether the machine translation or model outputs match the references exactly (100%). If it matches the references, the result is calculated by dividing the total number of correct outputs by the total number of references.
- **Word Error Rate (WER)**: WER uses edit distance to calculate the accuracy percentage. Instead of using the distance between phonemes, it uses the distance between words. The final result is calculated by dividing the total number of substitutions + deletions + insertions by the total number of words in the reference.
- **Expression Recognition Rates (ExpRate):** ExpRate uses edit distance to calculate the accuracy of the model. For each line with distance $= 0$, the total number of correct lines is increased by 1. The final result is the number of correct lines divided by the total number of lines.

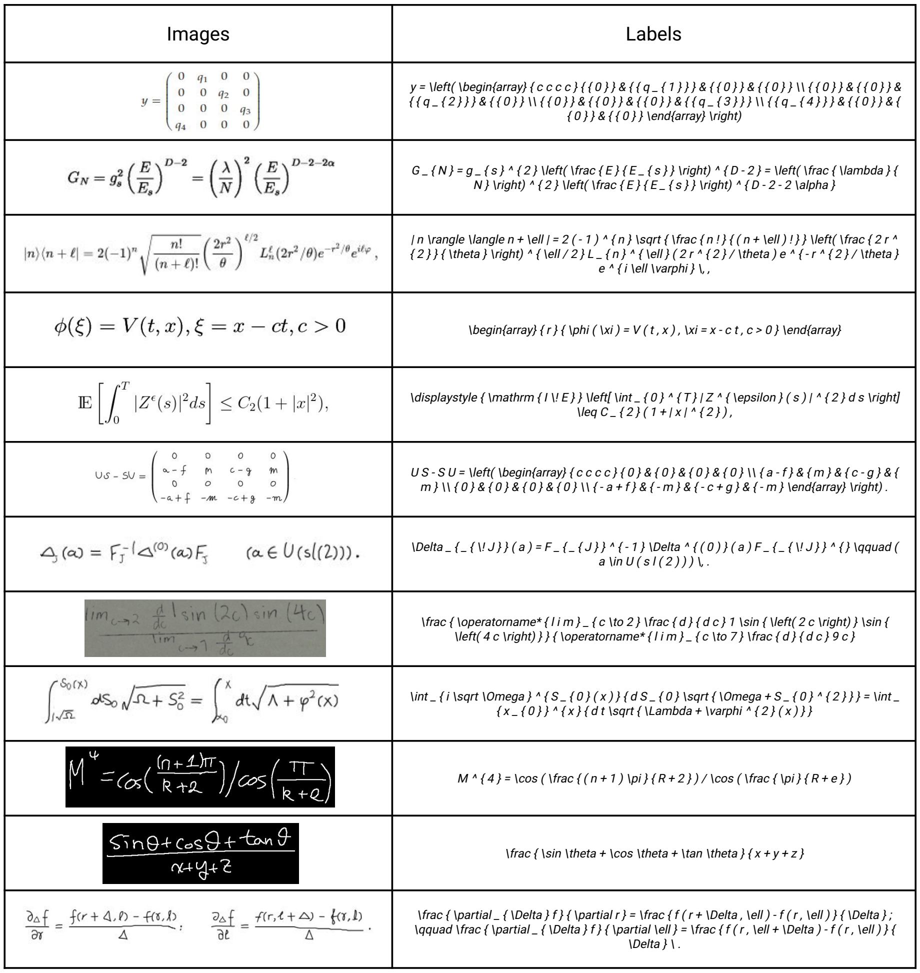
We use the benchmark Im2latex-100k dataset to conduct our experiments on printed mathematical expressions, which is a collection of about 100000 real-world mathematical expressions rendered from public papers on the arxiv.org server. This is a popular and dedicated dataset for the printed formula recognition task, and the test set contains 10285 samples. We use metrics such as BLEU score (4-gram), EM score, and ED score to evaluate on this dataset.

In addition, we use the benchmark CROHME dataset to demonstrate the effectiveness of the model on the handwritten formula recognition task, which is a large open dataset for handwritten mathematical expressions. The test set consists of three versions: CROHME 2014 (986 samples), CROHME 2016 (1147 samples), and CROHME 2019 (1199 samples). We choose expression level metrics: ExpRate (%), ExpRate ≤ 1 error (%), ExpRate ≤ 2 error (%), and ExpRate ≤ 3 error (%) provided by the CROHME 2019 organizers [17]. In addition, we use the WER metric to evaluate the error at the word level.

4.3 Implementation Details

The whole model is implemented using Pytorch & Transformer framework. We trained two versions: base (349m) and small (247m). Both versions use a max length of 512 and an image resolution of 224 × 768 (height × width). We use the Lion optimizer [7] with weight decay = 1e-2, $\beta_1 = 0.95$ and $\beta_2 = 0.98$. For the learning rate of 1e-4, we use 1k warmup steps based on the cosine schedule.

The model was trained for 6 epochs for both base and small with a batch size of 2048 with gradient accumulation on 1 GPU Tesla P100-PCIE-16GB. The total training time was about 50 days.



Images	Labels
$y=\begin{pmatrix}0&q_1&0&0\\0&0&q_2&0\\0&0&0&q_3\\q_4&0&0&0\end{pmatrix}$	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)
$G_N=g_s^2\left(\frac{E}{E_s}\right)^{D-2}=\left(\frac{\lambda}{N}\right)^2\left(\frac{E}{E_s}\right)^{D-2-2\alpha}$	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 }
$\vert n\rangle\langle n+\ell\vert=2(-1)^n\sqrt{\frac{n!}{(n+\ell)!}}\left(\frac{2r^2}{\theta}\right)^{\ell/2}L_n^\ell(2r^2/\theta)e^{-r^2/\theta}e^{i\ell\varphi}\,,$	| n \rangle \langle n + \ell | = 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 } \, ,
$\phi(\xi)=V(t,x),\xi=x-ct,c>0$	\begin{array} { r } { \phi (\xi) = V (t , x) , \xi = x - c t , c > 0 } \end{array}
$\mathbb{E}\left[\int_0^T\vert Z^\epsilon(s)\vert^2ds\right]\leq C_2(1+\vert x\vert^2),$	\displaystyle { \mathrm { I \! E } } \left[\int _ { 0 } ^ { T } | Z ^ { \epsilon } (s) | ^ { 2 } d s \right] \leq C _ { 2 } (1 + | x | ^ { 2 }) ,
$US-SU=\begin{pmatrix}0&0&0&0\\a-f&m&c-g&m\\0&0&0&0\\-a+f&-m&-c+g&-m\end{pmatrix}$	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) .
$\Delta_J(a)=F_J^{-1}\Delta^{(0)}(a)F_J\qquad(a\in U(sl(2))).$	\Delta _ { _ { \! J } } (a) = F _ { _ { J } } ^ { - 1 } \Delta ^ { (0) } (a) F _ { _ { \! J } } ^ { } \qquad (a \in U (s l (2))) \, .
$\frac{\lim_{c\to2}\frac{d}{dc}1\sin(2c)\sin(4c)}{\lim_{c\to7}\frac{d}{dc}9c}$	\frac { \operatorname* { l i m } _ { c \to 2 } \frac { d } { d c } 1 \sin { \left(2 c \right) } \sin { \left(4 c \right) } } { \operatorname* { l i m } _ { c \to 7 } \frac { d } { d c } 9 c }
$\int_{i\sqrt{\Omega}}^{S_0(x)}dS_0\sqrt{\Omega+S_0^2}=\int_{x_0}^{x}dt\sqrt{\Lambda+\varphi^2(x)}$	\int _ { i \sqrt \Omega } ^ { S _ { 0 } (x) } { d S _ { 0 } \sqrt { \Omega + S _ { 0 } ^ { 2 } } } = \int _ { x _ { 0 } } ^ { x } { d t \sqrt { \Lambda + \varphi ^ { 2 } (x) } }
$M^4=\cos\left(\frac{(n+1)\pi}{R+2}\right)/\cos\left(\frac{\pi}{R+e}\right)$	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 { \sin \theta + \cos \theta + \tan \theta } { x + y + z }
$\frac{\partial_\Delta f}{\partial r}=\frac{f(r+\Delta,\ell)-f(r,\ell)}{\Delta};\qquad\frac{\partial_\Delta f}{\partial\ell}=\frac{f(r,\ell+\Delta)-f(r,\ell)}{\Delta}.$	\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. 4: Data examples

Dataset	Model	ExpRate	$\leq$ 1 error	$\leq$ 2 error	$\leq$ 3 error
CROHME 14	DenseWAP [42]	43.0	57.8	61.9	-
	DenseWAP-TD [43]	49.1	64.2	67.8	-
	WS-WAP [33]	53.65	-	-	-
	Li *et al.* [20]	56.59	69.07	75.25	78.60
	Ding *et al.* [10]	58.72	-	-	-
	BTTR [46]	53.96	66.02	70.28	-
	BTTR (CoMER) [45]	55.17	67.85	72.11	74.14
	CoMER [45]	**59.33**	71.70	75.66	77.89
	PAL [39]	39.66	56.80	68.51	-
	WAP [44]	46.55	61.16	65.21	66.13
	PGS [17]	48.78	66.13	73.94	-
	PGS-v2 [38]	48.88	64.50	69.78	-
	DLA [15]	49.85	-	-	-
	ABM [4]	56.85	**73.73**	81.24	-
	WYGIWYS [8]	36.4	-	-	-
	Sumen-base (ours)	57.71	73.23	**81.54**	**86.21**
	Sumen-small (ours)	52.84	69.88	78.9	83.57
CROHME 16	DenseWAP [42]	40.1	54.3	57.8	-
	DenseWAP-TD [43]	48.5	62.3	65.3	-
	WS-WAP [33]	51.96	64.34	70.10	72.97
	Li *et al.* [20]	54.58	69.31	73.76	76.02
	Ding *et al.* [10]	57.72	70.01	76.37	78.90
	BTTR [46]	52.31	63.90	68.61	-
	BTTR (CoMER) [45]	56.58	68.88	74.19	76.90
	CoMER [45]	59.81	74.37	80.30	82.56
	WAP [44]	44.55	57.10	61.55	62.34
	PGS [17]	36.27	-	-	-
	PGS-v2 [38]	49.61	64.08	70.27	-
	DLA [15]	47.34	-	-	-
	ABM [4]	52.92	69.66	78.73	-
	Sumen-base (ours)	**76.02**	**85.53**	**92.33**	**94.59**
	Sumen-small (ours)	72.36	84.57	90.15	92.33
CROHME 19	DenseWAP [42]	41.7	55.5	59.3	-
	DenseWAP-TD [43]	51.4	66.1	69.1	-
	Ding *et al.* [10]	61.38	75.15	80.23	82.65
	BTTR [46]	52.96	65.97	69.14	-
	BTTR (CoMER) [45]	59.55	72.23	76.06	78.40
	CoMER [45]	62.97	77.40	81.40	83.07
	ABM [4]	53.96	71.06	78.65	-
	Sumen-base (ours)	**72.97**	**81.95**	**88.32**	**90.58**
	Sumen-small (ours)	69.49	81.17	86.57	88.67

Table 1: 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

The main experimental results are shown in Table 1 and Table 2 for the CROHME benchmark, and in Table 3 for the Img2latex-100k benchmark. We compare both the base and small versions with other methods, where the base has a larger architecture and performs better than the small.

For the handwritten formula recognition task evaluated on the CROHME dataset (Table 1), our model shines. It achieves impressive results, securing the top spot in 4 out of 4 categories on CROHME 16 & 19. However, on CROHME 14, the performance is slightly more nuanced. While we still achieve the best results in 2 out of 4 categories, the ExpRate metric places us in second place behind CoMER. Encouragingly, the overall performance suggests that our model surpasses existing solutions for handwritten formula recognition.

Dataset	Model	ExpRate	WER
CROHME 14	Dense [42]	50.1	13.9
	Dense+MSA [42]	52.8	12.9
	ABM [42]	56.85	10.1
	Sumen-base (ours)	**57.71**	**9.07**
	Sumen-small (ours)	52.84	10.97
CROHME 16	Dense [42]	47.5	15.4
	Dense+MSA [42]	50.1	13.7
	Sumen-base (ours)	**76.02**	**3.9**
	Sumen-small (ours)	72.36	5.7
CROHME 19	Sumen-base (ours)	**72.97**	**5.88**
	Sumen-small (ours)	69.49	6.98

Table 2: Comparison of performance handwritten formula recognition task with previous methods on CROHME 2014/2016/2019 test sets based on word error rate.

Moving on to printed mathematical expressions, the results are a touch more modest. When evaluating the entire string for accuracy (BLEU and ED scores), our model excels, achieving the highest marks. However, the EM score, which focuses on exact character-by-character precision, falls within the average range. Upon closer inspection, we identified instances where one or more characters were incorrectly recognized, leading to a score of 0 for the entire string in these specific cases. This highlights an area for further refinement in future iterations of the model.

Experimental results demonstrate that our model achieves high performance on both handwritten and printed mathematical expression recognition tasks. Overall, the model outperforms existing methods on the CROHME dataset for handwritten expressions and achieves the highest BLEU and ED scores for printed expressions. However, the EM score for printed expressions still needs to be improved in future versions.

Model	BLEU	Edit Distance	Exact Match
INFTY [32]	66.65	96.4	59.6
CNNENC [8]	75.01	61.17	53.53
WYGIWYS [8]	87.73	87.60	77.46
MI2LS w/o Reinforce [37]	89.08	91.09	79.39
MI2LS with Reinforce [37]	90.28	92.28	82.33
DoubleAttention [37]	88.42	88.57	79.81
DenseNet [36]	88.25	91.57	-
MathBERT [29]	90.45	90.11	**87.52**
Zhou *et al.* [47]	92.11	90.0	60.2
Wang *et al.* [36]	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)	**94.51**	**96.59**	61.81
Sumen-small (ours)	93.79	96.34	56.29

Table 3: Comparison of performance printed formula recognition task with previous methods on Im2latex-100k test set.

5 Conclusion

We successfully address the grand challenge of recognizing both printed and handwritten mathematical formulas in a single model. While previous studies struggled to handle complex formulas, our novel approach surpasses them by employing the Sumen architecture with improvements in the Vision module and a large-scale dataset to enhance the model's generalization capability. This outcome clearly demonstrates Sumen's superior performance compared to state-of-the-art methods. Specifically, BLEU and ED scores on the img2latex100k benchmark validate the model's diverse and accurate capabilities in recognizing and converting images to LaTeX format. Furthermore, achieving the highest WER and Exprate scores on the CROHME benchmark confirms Sumen's flexibility and effectiveness for various types of mathematical formulas. Conversely, although the EM metric is not the highest, BLEU and ED are the highest. This is because EM is used to evaluate individual characters, while BLEU and ED are used to evaluate the overall string. Incorrect characters in the code can lead to poor EM results.

References

[1] AlFrauch. `https://huggingface.co/datasets/AlFrauch/im2latex`.

[2] Azu. https://huggingface.co/datasets/Azu/Handwritten-Mathematical-Expression-Convert-LaTeX.

[3] Benjamin P. Berman and Richard J. Fateman. "Optical Character Recognition for Typeset Mathematics". In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC '94, Oxford, UK, July 20-22, 1994*. Ed. by Malcolm A. H. MacCallum. ACM, 1994, pp. 348–353. DOI: 10.1145/190347.190438. URL: https://doi.org/10.1145/190347.190438.

[4] 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.

[5] 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.

[6] Chungkwong Chan. "Stroke Extraction for Offline Handwritten Mathematical Expression Recognition". In: *IEEE Access* 8 (2020), pp. 61565–61575. DOI: 10.1109/ACCESS.2020.2984627.

[7] 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.

[8] 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.

[9] 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.

[10] Haisong Ding, Kai Chen, and Qiang Huo. "An Encoder-Decoder Approach to Handwritten Mathematical Expression Recognition with Multi-head Attention and Stacked Decoder". In: *16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, September 5-10, 2021, Proceedings, Part II*. Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. Vol. 12822. Lecture Notes in Computer Science. Springer, 2021, pp. 602–616. DOI: 10.1007/978-3-030-86331-9_39. URL: https://doi.org/10.1007/978-3-030-86331-9%5C_39.

[11] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7,*

2021. OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[12] gregoryeritsyan. https://www.kaggle.com/datasets/gregoryeritsyan/im2latex-230k.

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

[14] Geewook Kim et al. "OCR-Free Document Understanding Transformer". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVIII*. Ed. by Shai Avidan et al. Vol. 13688. Lecture Notes in Computer Science. Springer, 2022, pp. 498–517. DOI: 10.1007/978-3-031-19815-1_29. URL: https://doi.org/10.1007/978-3-031-19815-1%5C_29.

[15] 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, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 2413–2418. DOI: 10.1109/CVPRW50498.2020.00291. URL: https://openaccess.thecvf.com/content%5C_CVPRW%5C_2020/html/w34/Le%5C_Recognizing%5C_Handwritten%5C_Mathematical%5C_Expressions%5C_via%5C_Paired%5C_Dual%5C_Loss%5C_Attention%5C_Network%5C_CVPRW%5C_2020%5C_paper.html.

[16] Anh Duc Le, Bipin Indurkhya, and Masaki Nakagawa. "Pattern Generation Strategies for Improving Recognition of Handwritten Mathematical Expressions". In: *CoRR* abs/1901.06763 (2019). arXiv: 1901.06763. URL: http://arxiv.org/abs/1901.06763.

[17] 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.

[18] Hsi-Jian Lee and Jiumn-Shine Wang. "Design of a mathematical expression recognition system". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 2. 1995, 1084–1087 vol.2. DOI: 10.1109/ICDAR.1995.602097.

[19] Vladimir I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics. Doklady* 10 (1965), pp. 707–710. URL: https://api.semanticscholar.org/CorpusID:60827152.

[20] 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.

[21] Yinhan Liu et al. "Multilingual Denoising Pre-training for Neural Machine Translation". In: *Trans. Assoc. Comput. Linguistics* 8 (2020), pp. 726–742.

DOI: 10.1162/TACL_A_00343. URL: https://doi.org/10.1162/tacl%5C_a%5C_00343.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] OleehyO. https://huggingface.co/datasets/OleehyO/latex-formulas.

[27] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040/.

[28] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, 2013, pp. 1310–1318. URL: http://proceedings.mlr.press/v28/pascanu13.html.

[29] 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.

[30] rvente. https://www.kaggle.com/datasets/rvente/im2latex170k.

[31] 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.

[32] 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.

[33] 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.

[34] v7labs. https://www.v7labs.com/open-datasets/aida.

[35] 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.

[36] Jian Wang, Yunchuan Sun, and Shenling Wang. "Image To Latex with DenseNet Encoder and Joint Attention". In: *2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018, Beijing, China, October 19-21, 2018*. Ed. by Rongfang Bie, Yunchuan Sun, and Jiguo Yu. Vol. 147. Procedia Computer Science. Elsevier, 2018, pp. 374–380. DOI: 10.1016/J.PROCS.2019.01.246. URL: https://doi.org/10.1016/j.procs.2019.01.246.

[37] 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.

[38] 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.

[39] Jin-Wen Wu et al. "Image-to-Markup Generation via Paired Adversarial Learning". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*. Ed. by Michele Berlingerio et al. Vol. 11051. Lecture Notes in Computer Science. Springer, 2018, pp. 18–34. DOI: 10.1007/978-3-030-10925-7_2. URL: https://doi.org/10.1007/978-3-030-10925-7%5C_2.

[40] Richard Zanibbi, Dorothea Blostein, and James R. Cordy. "Recognizing Mathematical Expressions Using Tree Transformation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.11 (2002), pp. 1455–1467. DOI: 10.1109/TPAMI.2002.1046157. URL: https://doi.org/10.1109/TPAMI.2002.1046157.

[41] Jianshu Zhang, Jun Du, and Li-Rong Dai. "A GRU-based Encoder-Decoder Approach with Attention for Online Handwritten Mathematical Expression Recognition". In: *CoRR* abs/1712.03991 (2017). arXiv: 1712.03991. URL: http://arxiv.org/abs/1712.03991.

[42] 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.

[43] 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.

[44] 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.

[45] Wenqi Zhao and Liangcai Gao. "CoMER: Modeling Coverage for Transformer-Based Handwritten Mathematical Expression Recognition". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVIII.* Ed. by Shai Avidan et al. Vol. 13688. Lecture Notes in Computer Science. Springer, 2022, pp. 392–408. DOI: 10.1007/978-3-031-19815-1_23. URL: https://doi.org/10.1007/978-3-031-19815-1%5C_23.

[46] 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.

[47] 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.