Journal of Information Systems Engineering and Management
2025, 10(13s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Transformer-Based Image-to-LaTeX Conversion: Improving Mathematical Equation Recognition with Self-Attention

Dr. Neeta P. Patil1, Dr. Yogita D. Mane2, Akshay Agrawal3, Dr. Anil Vasoya4, Sanketi Raut5
1,4 Associate Professor, Dept. of IT, Thakur College of Engg. and Tech., Mumbai, Maharashtra, India
2 Associate Professor, Dept. of AI&DS, Thakur College of Engg. and Tech., Mumbai, Maharashtra, India
3,5 Assistant Professor, Dept. of IT, Universal College of Engineering, Mumbai, Maharashtra, India
**Corresponding email: npat78691@gmail.com*

ARTICLE INFO

Received: 28 Nov 2024
Revised: 09 Jan 2025
Accepted: 31 Jan 2025

ABSTRACT

Automating the transfer of mathematical equations from photos to LaTeX code is difficult because to handwriting diversity, formatting issues, and structural difficulties. Traditional CNN and RNN models struggle with long-term dependency and input variability. To overcome these challenges, we present a transformer-based encoder-decoder architecture that uses self-attention to increase contextual comprehension and sequence alignment. The model is trained on the im2latex dataset using token-level cross-entropy loss and sequence-level BLEU-based reinforcement learning, followed by Adam and beam search for inference. Compared to existing models, the proposed model has the greatest BLEU score, competitive MED performance, and better robustness against noisy and handwritten inputs, however the Exact Match (EM) score shows space for improvement. This study demonstrates the efficacy of transformer-based architectures for improving LaTeX conversion accuracy and mathematical document processing.

Keywords: Transformer model; encoder-decoder; image to LaTeX; math formulas

INTRODUCTION

Mathematical formulas play a pivotal role in numerous disciplines, including science, engineering, and mathematics. Converting these formulas into structured markup languages like LaTeX facilitates seamless integration into digital documents and enhances information retrieval capabilities. However, recognizing mathematical formulas from images poses significant challenges due to their complex spatial arrangements and hierarchical structures [1]. Traditional OCR-based methods often fail to accurately parse intricate relationships such as fractions, subscripts, superscripts, and matrices [2], [6].

Recent advances in deep learning, particularly sequence-to-sequence (seq2seq) models, have demonstrated promising potential in addressing these challenges [3], [5]. These models typically utilize CNNs for feature extraction and RNNs for sequence generation. Despite their effectiveness, such architectures encounter difficulties with modeling long-range dependencies and suffer from computational inefficiencies [8]. To overcome these limitations, this study proposes a transformer-based approach leveraging the im2latex dataset. Transformers, equipped with powerful attention mechanisms, are adept at modeling global dependencies, enabling precise and scalable translation of mathematical formulas.

The transformer-based model represents an innovative solution for converting images of mathematical equations into LaTeX code. By employing the im2latex dataset, the system achieves state-of-the-art performance in understanding and translating intricate mathematical notations [12]. This document provides a comprehensive overview of the methodology, training procedures, and key advantages of this approach.

The process of transforming mathematical equations from image formats to LaTeX code involves addressing both spatial and sequential complexities [1]. Unlike traditional OCR-based systems [8], the transformer-based model excels in capturing spatial relationships and generating accurate textual representations, thanks to its reliance on attention mechanisms. The im2latex dataset serves as a robust foundation for training this model, offering real-world mathematical expressions paired with their corresponding LaTeX markup.

RELATED WORKS

A computer vision-based LaTeX editor [6] categorizes recognition methods into online and offline approaches, detecting pen tip movements within image frames to delineate character bounding boxes, which are then processed using chain extraction techniques. However, variations in handwriting styles and character ambiguity lead to recognition errors, necessitating the integration of contextual knowledge for improved accuracy.

An encoder-decoder model with an attention mechanism [10] maps images of mathematical formulas to LaTeX markup, effectively addressing the Im2Latex problem by enhancing the neural encoder-decoder framework with visual attention. Despite achieving over 70% alignment between predicted and target images, the model generates incorrect sequences in approximately 40% of cases. To improve the efficiency of LaTeX equation generation, an automated system [7] converts images of mathematical formulas into LaTeX code, utilizing optimization techniques such as Stochastic Gradient Descent, Adam, and RMSprop, along with weight initialization methods like Glorot Uniform and He Normal. However, the system demonstrates limited accuracy when processing conventional screenshots of mathematical equations.

A deep learning-based approach [9] automates handwritten equation conversion into LaTeX, leveraging neural networks, though accuracy remains a concern due to the complexities of emerging technologies. A versatile deep learning-based system [11] is introduced for decompiling images into presentational markup without requiring prior knowledge of LaTeX. It integrates a convolutional neural network for text and layout recognition with an attention-based neural machine translation framework and presents two novel datasets—one comprising real-world mathematical expressions with LaTeX markup and another containing synthetic web pages with HTML snippets, increasing the complexity of the training process. Another model [14] employs a multi-layer convolutional network for feature extraction combined with an attention-based RNN decoder, introducing a novel source encoder layer for OCR tasks. It utilizes a multi-row recurrent model within the encoder to capture document layouts, demonstrating that neural OCR can operate without the left-to-right ordering assumptions typical of CTC-based models. Additionally, it highlights the critical role of fine-grained attention in achieving high recognition accuracy.

The image-to-markup generation problem is explored by Deng et al. [16], who compiled a dataset of approximately 100,000 mathematical formulas extracted from LaTeX sources in arXiv papers. The formulas, rendered using pdflatex and converted into PNG images, were used to train a CNN-RNN model with an attention mechanism. However, a rigid downsampling approach limits the model's flexibility and performance on diverse input data. The pix2tex tool [13] facilitates LaTeX code extraction from images through both a command-line interface and a graphical interface invoked via the latexocr command. While it streamlines equation recognition by allowing users to capture screenshots and generate LaTeX code rendered with MathJax, its performance declines when handling larger images.

A CNN-based model [17] automates LaTeX equation recognition by leveraging high-quality datasets, inspired by neural network success in handwritten digit recognition. However, maintaining precision in longer sequences remains a challenge as the model struggles to accurately predict characters positioned further from the initial ones. Finally, the "Math Utilities" tool [15] enables the conversion of mathematical content between Microsoft Word and LaTeX, providing a versatile solution for translating across multiple formats. However, due to the extensible nature of LaTeX, designing a universal translator that accommodates all possible variations and edge cases remains a significant challenge.

Several well-established LaTeX conversion systems offer efficient solutions for document formatting and cross-platform compatibility:

- Pandoc [19]: Pandoc is a highly versatile and robust document conversion tool, supporting a broad spectrum of input and output formats, including LaTeX. It facilitates seamless translation between formats such as Markdown, HTML, Word, and others, positioning it as an indispensable tool for interdisciplinary document workflows.
- TeX4ht [20]: TeX4ht is a powerful LaTeX to HTML/XML converter that also supports the conversion of LaTeX documents into formats like OpenOffice, DocBook, and RTF. With its extensive compatibility with numerous LaTeX packages and commands, it excels in handling complex documents, making it a preferred choice for users dealing with multifaceted LaTeX content.

- Latex2rtf [21]: Latex2rtf is a LaTeX to RTF converter that enables the generation of high-quality Word documents from LaTeX source files. Supporting most LaTeX commands and packages, it produces accurate output, making it particularly valuable for users needing to integrate LaTeX content into Microsoft Word.
- MathJax [22]: MathJax is a sophisticated JavaScript library for rendering mathematical equations and symbols in web pages. It converts LaTeX code into MathML or HTML+CSS, providing an elegant solution for embedding complex mathematical expressions in online documents and web applications.

While these LaTeX conversion tools significantly enhance efficiency, it is important to recognize that they may not always yield perfect results. In some cases, manual refinement may be required to ensure the quality and precision of the final output.

A thorough analysis of the current literature finds various persistent issues with picture-to-LaTeX conversion methods, including image resolution, character spacing, and formatting discrepancies. One key challenge is recognition failures due by irregular handwriting and the ambiguity of similar-looking letters, which can be addressed using contextual information and transformer-based models that can handle varied input forms. Furthermore, encoder-decoder models with attention mechanisms frequently produce incorrect sequences, resulting in improper LaTeX outputs. Conventional screenshots of mathematical equations also present difficulties, as they frequently result in incorrect predictions due to insufficient preprocessing. Furthermore, a disparity in distribution among preprocessed dataset images and real-world inputs from users may hinder model performance [3].

A streamlined training strategy addresses this issue by increasing adaptability to various input formats. Another issue to consider is the accuracy of models that use emerging technologies, which can be enhanced by rigorous refining and thorough training on real-world datasets.

Parsing LaTeX markup adds complexity, as syntax ambiguities can lead to errors. Advanced parsing approaches can help to mitigate such difficulties. Furthermore, many current systems lack a user-friendly interface for entering mathematical formulas, reducing accessibility. A well-designed web-based interface that includes an interactive input canvas improves usability, making the system more efficient and user-centric [11], [15]. Addressing these issues together results in a more accurate, reliable, and user-friendly image-to-LaTeX conversion system.

Transformer-based models efficiently handle critical issues in image-to-LaTeX conversion [18]. Their self-attention techniques improve character identification by collecting contextual dependencies and lowering errors due to confusing symbols. Unlike classic models, transformers use global attention to improve sequence generation and reduce inaccuracies. Vision transformers (ViTs) improve screenshot inputs by extracting robust features from variable-resolution photos. They also adapt effectively to different input styles, which helps to mitigate performance concerns caused by distribution shifts. Advanced parsing mechanisms eliminate LaTeX syntax errors, and seamless interaction with web-based tools improves the user experience. Overall, transformers offer a more precise, durable, and user-friendly solution.

PROPOSED SYSTEM

The process of converting mathematical equations and tables from various sources into LaTeX format can be time-consuming and error-prone. Many researchers, students, and professionals in fields such as mathematics, physics, and engineering require a tool that can easily convert equations and tables to LaTeX format. The current solutions available for converting equations and tables to LaTeX format require manual copying and pasting, which can lead to errors and inconsistencies. Furthermore, there are often formatting issues that arise during the conversion process that require manual intervention. Therefore, we have come up with an idea to develop a tool that can accurately and efficiently convert mathematical equations and tables from various sources into LaTeX format. The tool should be able to handle equations and tables of various complexities and formats, including images and scanned documents. The converter should also include a user-friendly interface and be able to produce high-quality LaTeX output that is free of errors and formatting issues. The development of an accurate and efficient Equation to LaTeX and Table to LaTeX Converter will greatly benefit researchers, students, and professionals by saving time and reducing errors during the conversion process [16].

The "Equation to LaTeX" and "Table to LaTeX" projects both aim to convert mathematical expressions and tables respectively into LaTeX format, which is widely used in scientific and academic publishing.

Equation:

➢ Equation can be scanned to obtain LaTeX code.

- ➢ Equation's image can be uploaded of .jpg, .jpeg or .png format to obtain LaTeX code.

Table:

- ➢ Table can be uploaded in .pdf format to obtain its LaTeX equivalent code.
- ➢ Table in excel file needs to be saved as pdf to upload and generate LaTeX code.

LaTeX is a popular markup language used in many scientific and academic fields, and the output generated by these tools can be easily integrated into documents written in LaTeX. By converting equations and tables into LaTeX format, it becomes easier for researchers and students to access and use the information for their own work.

The conversion process can be time-consuming and error-prone when done manually. Using an automated tool like Equation to LaTeX or Table to LaTeX can significantly improve the efficiency of the process. Automated tools can ensure that the output is consistent and accurate across multiple equations and tables.

Transformer Model Architecture:

The proposed model consists of two primary components: a Vision Transformer (ViT) encoder and a transformer decoder as in Fig.1. Together, these components form an end-to-end trainable system for image-to-LaTeX translation.

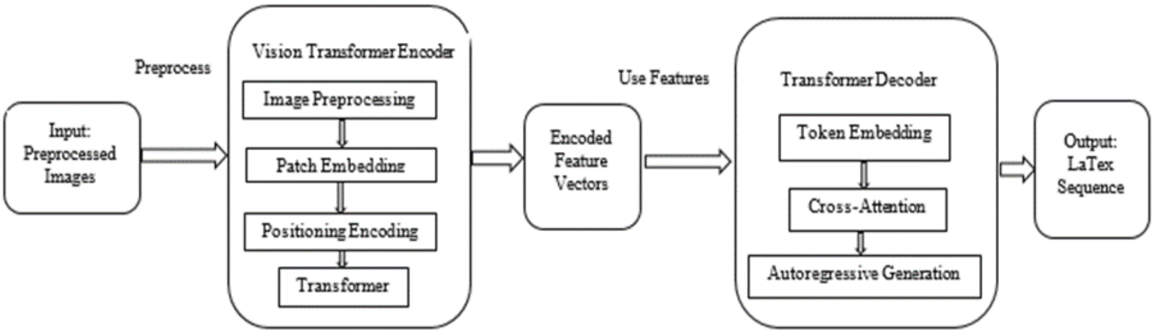

Fig. 1. The proposed encoder-decoder architecture of the transformer model.

Vision Transformer Encoder: The Vision Transformer (ViT) encoder is designed to extract both spatial and contextual features from input images. It operates through the following steps:

Image Preprocessing: Input images are resized to a fixed resolution (e.g., 256×256) and divided into non-overlapping patches (e.g., 16×16 pixels). The patches are then flattened into vectors.

- Patch Embedding: These flattened patches are linearly transformed into embedding vectors of a fixed dimension to represent local image features numerically.
- Positional Encoding: Positional information is added to the patch embeddings, ensuring the retention of spatial relationships and structural locality within the image.
- Transformer Layers: The sequence of patch embeddings is passed through multiple transformer encoder layers, each incorporating multi-head self-attention and feed-forward networks. This process generates a globally contextualized representation of the input image.

Transformer Decoder: The transformer decoder generates LaTeX sequences in an autoregressive manner, leveraging the encoded image features. Its operation is described as follows:

- Token Embedding: LaTeX tokens are mapped into a continuous vector space, enabling the model to work with numerical representations of text.
- Cross-Attention: The decoder aligns the encoded image features with the LaTeX tokens by using a cross-attention mechanism, ensuring accurate mapping between visual regions and textual tokens.
- Autoregressive Generation: Tokens are predicted sequentially, with each prediction conditioned on previously generated tokens and the encoder's output. Positional encodings are added to maintain the correct order of the sequence.

Workflow of the Transformer-Based Model

1. **Input**: Preprocessed images of mathematical formulas.
2. **Encoding**: Vision Transformer encodes the image into feature vectors.
3. **Decoding**: Transformer decoder generates LaTeX tokens using cross-attention with the image features.
4. **Output**: A complete LaTeX sequence representing the input equation.

Transformer-based models provide key advantages in image-to-LaTeX conversion. They effectively capture long-range dependencies using multi-head self-attention, enabling the modeling of relationships across image regions. Their scalability allows for efficient handling of large datasets and complex inputs. Cross-attention ensures accurate mapping between image features and LaTeX tokens, while the system's adaptability makes it robust to variations in input formats, including noisy or handwritten data.

LaTeX Table Conversion Libraries [23]: For converting tabular data into LaTeX format, several sophisticated Python libraries can streamline the process:

tabulate: A highly flexible library that offers an array of table formatting options, including LaTeX. It provides an efficient way to convert tabular data into LaTeX tables, suitable for a variety of formatting requirements.

pylatex: A robust Python library designed for generating LaTeX documents programmatically. It offers fine-grained control over document structure, making it ideal for users who need to create complex LaTeX tables with precise formatting.

excel2latex: A LaTeX package specifically designed for importing data from Microsoft Excel files directly into LaTeX tables. It supports various formatting options, including table alignment, column formatting, and style customization.

These libraries greatly simplify the conversion of tables into LaTeX, providing flexibility in choosing the most appropriate tool based on the specific needs of the data and the desired output format.

IMPLEMENTATION DETAILS

Dataset: The model is trained and evaluated on the im2latex dataset, which comprises over 100,000 pairs of images and corresponding LaTeX sequences. The dataset is split into training, validation, and test sets. The im2latex dataset is specifically curated for the image-to-LaTeX generation task. It consists of:

1. Images: Grayscale PNG images of mathematical formulas rendered using LaTeX tools like pdflatex.
2. LaTeX Code: Corresponding LaTeX markup for each formula.
3. File Structure:
   - formula_images/: Directory containing rendered images.
   - im2latex_formulas.lst: File listing LaTeX formulas.
   - Train, validation, and test splits for systematic evaluation.
4. Size: Over 100,000 pairs of images and LaTeX formulas.

This dataset supports tasks like sequence-to-sequence learning, image feature extraction, and LaTeX generation.

Preprocessing

1. Images are resized to a fixed resolution (e.g., 64x256).
2. Pixel values are normalized to [0, 1].
3. LaTeX formulas are tokenized, and sequences are padded for uniformity.

The input image as in Fig. 2 is given as input to the system.

$$R(\mathrm{e}^{\mathrm{i}\alpha}) = \mathrm{e}^{2\mathrm{i}\alpha} R(Z),$$

Fig. 2 Input Equation Image

Output Image

Fig.3 LaTeX Output

The above Fig. 3 shows the output which can be stored in the following ways:

- Copy to clipboard.
- Download as a word file.

The method for converting tables is also implemented.

The process of parsing tables from input text as in Fig. 4 and converting their content into LaTeX format requires a structured approach. Initially, the rows and columns of the table must be accurately detected and mapped. Following this, LaTeX templates are employed to generate a tabular environment that replicates the table's structure with precise formatting. This ensures that the resulting LaTeX code preserves the original alignment, spacing, and overall organization, resulting in a seamless and well-formatted LaTeX table representation.

Name	Roll No.
Lavanya D.	30
Maliha Momin	35
Divyanshu Singh	62
Neha Suratkal	68

Fig. 4 Input Table

Ways to take Table as input:

For seamless conversion of tables to LaTeX code, the following input methods can be employed:

- Direct PDF Upload: Tables can be provided in .pdf format, allowing for direct parsing and conversion into LaTeX code.
- Excel to PDF Conversion: Tables designed in Excel should first be exported as PDFs, which can then be uploaded for subsequent LaTeX code generation.

These approaches ensure that table content is accurately captured and converted into a structured and well-formatted LaTeX representation.

The resulting LaTeX code is compiled into a comprehensive output document, which can be saved as a .tex file or directly rendered into formats such as PDF.

Text Parsing: The process begins by parsing the input text to discern its structural components, including headings, paragraphs, lists, equations, and citations. This can be accomplished through natural language processing (NLP) techniques, regular expressions, or a combination of both, ensuring precise identification and extraction of elements.

After parsing, tailored LaTeX templates are applied to represent each component appropriately. For instance, section headings are formatted using \section{Heading text}, while lists are encapsulated within \begin{itemize} ... \end{itemize}. These templates can either be pre-established or generated dynamically, depending on the structure of the input. This approach ensures the creation of a well-organized, syntactically accurate LaTeX document that faithfully reflects the original content.

Generated Latex

```
\begin{tabular}{llr}
\toprule
{} &              Name &  Roll No. \\
\midrule
0 &        Lavanya D. &        30 \\
1 &      Maliha Momin &        35 \\
2 &  Divyanshu Singh &        62 \\
3 &     Neha Suratkal &        68 \\
\bottomrule
\end{tabular}
```

Fig. 5 LaTex Output

The above Fig. 5 shows the output which can be stored in the following ways:

- Copy to clipboard.
- Download as a word file.

Algorithm Overview: The algorithm delineates a comprehensive and sophisticated approach for processing images of mathematical equations and converting them into accurately structured LaTeX code, leveraging deep learning techniques for precision and reliability.

<u>General Workflow</u>

Step 1: Initiate Process

Step 2: Collect and preprocess the dataset.

Step 3: Construct and configure a Transformer-based model.

Step 4: Train the model using the prepared dataset.

Step 5: Validate and test the trained model.

Step 6: Display the generated LaTeX code corresponding to input images.

Step 7: Terminate Process

<u>Detailed Processing Steps</u>

Step 1: Initiate Process

Step 2: Upload an image containing a mathematical equation.

Step 3: Apply a Transformer-based feature extraction network to analyse and encode the image data.

Step 4: Extract textual content from the encoded image representation.

Step 5: Tokenize the extracted text sequence for structured processing.

Step 6: Detect and identify mathematical symbols within the tokenized sequence.

Step 7: Replace the detected mathematical symbols with their LaTeX code equivalents.

Step 8: Display the complete and formatted LaTeX code to the user.

Step 9: Terminate Process

In a transformer-based image-to-LaTeX model, training optimizes both token-level and sequence-level accuracy. Cross-entropy loss is minimized during initial training with teacher forcing to align predicted and ground truth tokens, followed by BLEU-based reinforcement learning to enhance global sequence quality. The Adam optimizer with learning rate scheduling and gradient clipping ensures stable convergence. The architecture comprises six transformer layers in both the encoder and decoder, an embedding dimension of 512, and a patch size of 16×16. The model is trained on 80% of the im2latex dataset, with 10% used for validation and testing. During inference, beam search selects the most probable LaTeX sequence, leveraging self-attention for contextual coherence.

RESULTS

Performance Analysis

Formula recognition can be viewed as a machine translation task, where the input consists of a sequence of trajectory points representing a formula, and the output is its corresponding LaTeX sequence. To evaluate the model's accuracy, commonly used machine translation metrics are applied [4].

BLEU (Bilingual Evaluation Understudy) measures how closely the generated output matches a human reference translation. Maximum Edit Distance (MED) determines the similarity between two sequences by calculating the number of modifications needed to make them identical. Exact Match (EM) checks whether the predicted LaTeX sequence perfectly matches the ground truth.

A comparison of this model with similar models is shown in Table 1. The results indicate that this model achieves a BLEU score about 2% higher than others, while its MED performance is comparable. However, the EM score has room for improvement.

Table 1: Comparison of similar models with the proposed models

Model	BLEU	MED	Exact Match (EM)
INFTY	67.10	54.30	16.20
WYGIWYS	87.90	87.80	77.10
DenseNet	88.40	91.30	-
DoubleAttention	88.60	88.75	79.50
MathBERT	90.50	90.25	87.30
MI2LS	90.15	91.75	82.10
Proposed Model	92.00	90.05	60.50

CONCLUSION

The proposed transformer-based model offers a robust and scalable solution for the translation of mathematical formulas into LaTeX, addressing a longstanding and complex challenge. By integrating a Vision Transformer encoder with a transformer decoder, the model achieves state-of-the-art performance on the im2latex dataset. The inclusion of attention mechanisms facilitates precise alignment between visual features and LaTeX tokens, effectively mitigating challenges such as exposure bias and intricate structural relationships. This innovative approach not only demonstrates exceptional accuracy and robustness but also establishes a foundation for further advancements. Future research could focus on augmenting the system with larger training datasets, sophisticated pretraining techniques, and support for multilingual LaTeX generation. Such enhancements would broaden its applicability and pave the way for transformative applications in scientific document digitization and formula recognition.

By leveraging the complementary strengths of Vision Transformers and transformer decoders, this model provides a powerful and efficient solution for converting mathematical images into LaTeX code. Its high accuracy and adaptability make it an invaluable tool across academic, scientific, and technical domains, solidifying its role as a cutting-edge technology in the field of mathematical formula recognition.

REFERENCES

[1] P. Ion, R. Miner, S. Buswell, and A. Devitt, *Mathematical Markup Language (MathML) 1.0 Specification*. World Wide Web Consortium (W3C), 1998.

[2] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep structured output learning for unconstrained text recognition," *arXiv preprint arXiv:1412.5903,* 2014.

[3] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in P*roceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156-3164.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311-318: Association for Computational Linguistics.

[5] Wang, Z., Liu, JC. Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training. *IJDAR* **24**, 63–75 (2021). https://doi.org/10.1007/s10032-020-00360-2

[6] Ozcan Oksiiz, Ujur Giidiikbay and Enis Getin, "COMPUTER VISION BASED TEXT AND EQUATION EDITOR FOR LATEX", IEEE International Conference on Multimedia and Expo (ICME), pp. 1255-1258, 2004.

[7] Sumeet S. Singh, "Teaching Machines to Code: Neural Markup Generation with Interpretable Attention", Independent Research, pp. 1-14, June 2018.

[8] Orji, E. Z., Haydar, A., Erşan, İ., & Mwambe, O. O. (2023). Advancing OCR Accuracy in Image-to-LaTeX Conversion—A Critical and Creative Exploration. *Applied Sciences*, *13*(22), 12503. https://doi.org/10.3390/app132212503

[9] Zelun Wang and Jyh-Charn Liu. 2020. PDF2LaTeX: A Deep Learning System to Convert Mathematical Documents from PDF to LaTeX. In Proceedings of the ACM Symposium on Document Engineering 2020 (DocEng '20). Association for Computing Machinery, New York, NY, USA, Article 4, 1–10. https://doi.org/10.1145/3395027.3419580

[10] Zhou, Mingle; Cai, Ming; Li, Gang; Li, Min, " An End-to-End Formula Recognition Method Integrated Attention Mechanism. Mathematics (2227-7390), 2023, Vol 11, Issue 1, p177, DOI 10.3390/math11010177

[11] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling and Alexander M. Rush, "Image-to-Markup Generation with Coarse-to-Fine Attention", Jun 2017.

[12] King Yiu Suen, "Image to LaTeX", Available [Online]: "https://github.com/kingyiusuen/image-to-latex", Accessed date: Nov 2022.

[13] Lukas Blecher, "LaTeX-OCR", Available [Online]: "https://github.com/lukas-blecher/latex-ocr", Accessed date: Nov. 2022.

[14] Avinash More, "IMAGE TO LATEX VIA NEURAL NETWORKS", Master' s Projects. 602, 2018.

[15] W. Ross Stone, "Converting between TEX or LaTeX and Word", IEEE Antennas & Propa gation Magazine, pp. 121-123, Aug 2017.

[16] Jiang, N., Liang, S., Wang, C., Wang, J., & Tan, L. (2024). LATTE: Improving Latex Recognition for Tables and Formulae with Iterative Refinement. *ArXiv, abs/2409.14201*.

[17] Zhang, Y., & Li, G. (2024). Improving Handwritten Mathematical Expression Recognition via Integrating Convolutional Neural Network with Transformer and Diffusion-Based Data Augmentation. *IEEE Access, 12*, 67945-67956.

[18] Pang, N., Yang, C., Zhu, X., Li, J., & Yin, X. (2021). Global Context-Based Network with Transformer for Image2latex. *2020 25th International Conference on Pattern Recognition (ICPR)*, 4650-4656.

[19]"Pandoc a universal document converter", Available [Online]: "https://pandoc.org/MANUAL.html", Accessed on: Jul 2024.

[20] TeX4ht, "LaTeX packages and custom commands", Available [Online]: "https://tug.org/tex4ht/", Accessed on: Jul 2024.

[21] "latex2rtf", Available [Online]: "https://latex2rtf.sourceforge.net/", Accessed on: Aug. 2024.

[22] "MathJax", Available [Online]: "https://docs.mathjax.org/en/latest/basic/mathjax.html", Accessed on: Aug. 2024.

[23] "Mathpix", Available [Online]: " https://mathpix.com/table-to-latex", Accessed on: Aug. 2024.