# Improving on Image to LaTeX OCR

**Griffin Bacheldor**
COMP 755
Chapel Hill, NC 27599-2200
gtbachel@unc.edu

**Anubhav Garg**
COMP 755
Chapel Hill, NC 27599-2200
anubhavg@cs.unc.edu

**Advait Parmar**
COMP 755
Chapel Hill, NC 27599-2200
advaitp@cs.unc.edu

**Om Shewale**
COMP 755
Chapel Hill, NC 27599-2200
oshewale@unc.edu

## Abstract

Converting equation images into structured, machine-readable LaTeX formats is critical for academia, education, and technical documentation. Existing solutions falter under real-world conditions like poor lighting, low-quality images, and handwritten equations. To address these challenges, we developed a robust and generalizable equation-to-LaTeX conversion system using a hybrid approach that combines Optical Character Recognition (OCR) with Natural Language Processing (NLP). Our model leverages Vision Transformers (ViT) for visual feature extraction and decoder LLMs for LaTeX generation, incorporating innovations such as preprocessing for noisy inputs, synthetic datasets for handwritten equations, and support for multi-equation systems. Our approach, evaluated primarily using BLEU scores, demonstrates significant robustness improvements through preprocessing and augmentation. By integrating cutting-edge techniques and optimizations like sparse quantization, our system advances the real-world applicability of automated mathematical transcription while remaining scalable and efficient.

## 1 Introduction

### 1.1 Background and Related Work

Advancements in deep learning have greatly enhanced the accuracy of image-to-LaTeX conversion systems. Deng et al. introduced an attention-based image-to-markup model that demonstrated promising results in converting rendered LaTeX equations back to their source code [18] Building on this work, Wang and Liu developed a deep learning system called PDF2LaTeX, which specifically targets the conversion of mathematical documents from PDF to LaTeX format [19] Converting equation images into structured, machine-readable formats like LaTeX is an increasingly vital task in many domains. Academic publishing, educational technology, and technical documentation depend heavily on accurate transcription of equations for readability, accessibility, and reproducibility. Yang et al. highlighted the importance of effective binarization methods for processing degraded documents, which is particularly relevant for handwritten equation recognition [20]. With the rapid growth of digital content, particularly scientific and mathematical materials, automating this process has become both a necessity and a challenge. Beyond convenience, automated transcription drastically reduces the time and effort required for manual entry while minimizing errors inherent to human transcription. However, the effectiveness of current equation-to-LaTeX conversion solutions is limited by several factors. State-of-the-art models excel in controlled environments, such as clean, high-resolution images captured under optimal lighting conditions. Yet, real-world applications often deal with less-than-ideal conditions. Handwritten equations, poor lighting, motion blur, skewed perspectives, and other distortions common in everyday image captures present significant challenges to existing systems. These limitations hinder the broader adoption and usability of automated tools in non-ideal scenarios, such as classrooms.

## 1.2 Project Motivation

Our project aims to address these shortcomings by developing a more robust and generalizable equation-to-LaTeX conversion system. Zhang et al. demonstrated the vulnerability of OCR systems to adversarial attacks, emphasizing the need for robust models that can handle various types of input distortions [21]. Our approach focuses on integrating advanced machine learning techniques, particularly transformer-based architectures. Transformer models have transformed fields like NLP and computer vision, offering promising results for OCR tasks. We propose a hybrid model combining Optical Character Recognition (OCR) with NLP to accurately interpret mathematical content from images. This dual-model approach leverages the Vision Transformer (ViT) for visual feature extraction and a language transformer such as RoBERTa for the syntactic generation of LaTeX. Together, these components offer a deeper understanding of both the visual and textual elements of equations, ensuring more accurate and context-aware output.
To address real-world challenges, we apply preprocessing techniques like contrast adjustment, perspective correction, and noise reduction.

### 1.2.1 Preprocessing Techniques

To simulate the challenges of real-world data, we employ pre-processing methods such as contrast adjustment, perspective correction, and noise reduction. These techniques prepare our model for handling diverse inputs, including low-quality photos. Following the work of Orji et al., we augment our training data with LaTeX syntax constraints to improve the model's understanding of mathematical notation structure. [22]

### 1.2.2 Handwritten Equation Support

A significant gap in existing models is their inability to process handwritten equations effectively. Using synthetic datasets of handwritten mathematical symbols, our model adapts to diverse handwriting styles, ensuring broader applicability.

### 1.2.3 Multi-Equation and Table Recognition

Many real-world documents contain not just standalone equations but also systems of equations and tables. Expanding our model's capability to handle these complexities aligns with our goal of creating a comprehensive transcription tool

## 1.3 Evaluation Metrics

To quantify performance, we rely primarily on the Google BLEU Score, which measures the similarity between the generated LaTeX code and the ground truth. The BLEU score is widely used in machine translation and provides an effective measure of precision across n-grams. Although BLEU gives us a solid benchmark for evaluating model outputs, it has some limitations in capturing structural mismatches or subtle variations in LaTeX code. Given more time, we would explore the incorporation of String Output Matching, a more robust metric capable of evaluating the structural and semantic fidelity of the generated LaTeX code. This metric is particularly valuable for assessing complex mathematical expressions in which minor errors could lead to entirely different interpretations. However, due to time constraints, our focus has remained on BLEU as the primary evaluation metric for this stage of the project.

## 1.4 Broader Impact

The integration of modern machine learning methods, such as sparse quantization and knowledge distillation, allows our system to balance high performance with efficiency. This scalability ensures its applicability across devices ranging from powerful workstations to laptops. Building on existing models like Im2Latex and TexTeller, our aim is to create a next-generation tool that pushes the boundaries of what is possible in automated equation recognition. Our project aims at a future where the equation-to-LaTeX conversion is seamless, accessible, and reliable, even under challenging conditions. By addressing the current limitations of OCR and NLP-based models, our goal is to create a solution that significantly improves workflows in academia, education, and beyond.

## 2 Related Works

### 2.1 Traditional OCR models

The advent of deep learning has significantly advanced OCR, with traditional methods relying on CNNs for feature extraction and RNNs for sequential text generation [2]. Traditional methods have usually relied on CNNs for feature extraction from images and RNNs for sequential text generation[3,4,5]. While effective in their own rights, these approaches often require additional language and vision models to improve accuracy along with involving sophisticated pre/post processing steps.

### 2.2 Transformer based OCR models

TrOCR enhances conventional OCR with a Transformer-based architecture, combining a Vision Transformer (ViT) encoder for visual features and a textual transformer decoder for text generation. It incorporates a Vision Transformer (ViT) [6] in the encoder to capture visual features and a textual Transformer, such as RoBERTa [7], in the decoder to generate text. This end-to-end method takes advantage of pre-trained models, resulting in improved text recognition performance [14].

### 2.3 Swin

The Swin Transformer [8] improves on ViTs by integrating hierarchical feature maps and window shifting, capturing both local and global image contexts. These features help capture global dependencies, reducing issues like occlusion or symbol misalignment in formula-heavy images. This is particularly valuable in scenarios where formulas span multiple lines or include nested structures [15]. This makes it a crucial model for processing images with complex characters, such as those containing LaTeX formulas. This makes it a crucial model for processing images with complex characters, such as those containing LaTeX formulas.

### 2.4 SwinV2

The SwinV2 model [9] improves upon the original Swin architecture with a more efficient design for hierarchical feature representation and window-based attention. It incorporates adaptive mechanisms for various input sizes, enhanced weight initialization, and mixed precision training, allowing it to optimize performance while managing model size. This larger architecture strikes a balance between complexity and efficiency, making SwinV2 a powerful and versatile model for OCR.

### 2.5 GPT-2 and BART

GPT-2 [10] and BART [11] are both powerful open-source models celebrated for their ability to generate coherent and contextually relevant text, making them excellent choices for decoders in our model. Notably, BART's bidirectional architecture and denoising pre-training significantly enhance its capabilities, enabling it not only to generate text but also to excel in understanding, summarization, and generalization. This unique design allows BART to leverage the contextual information produced by the encoder more effectively, resulting in richer and more nuanced outputs.

### 2.6 DistilGPT and TinyBERT

To optimize training and inference efficiency, we leverage compact variants of GPT-2 and BERT, specifically DistilGPT [16] and TinyBERT [17]. These models are created through Knowledge Distillation [18], which enables them to achieve significantly faster performance and reduced memory usage. However, this approach entails certain trade-offs in accuracy. Our study analyzes the implications of utilizing these lightweight models in the context of image-to-LaTeX conversion.

### 2.7 Existing Models

To address the task of converting mathematical formulas to latex code, several models have been developed, namely, Im2Latex, Pix2Text, and TexTeller and Sumen-base. Im2Latex and Pix2Text both follow vision encoder-decoder based architecture. Pix2Text and Texteller both a adopt an architecture similar to TrOCR with a few notable differences. Pix2Text uses ResNet as the encoder and a tranformer decoder, whereas TexTeller employs a Vision based Transformer(ViT) as the encoder and a transformer decoder. Sumen and Im2Latex follow a similar approach to ours, wherein they use a Swin transformer as an encoder, and a GPT-2 based decoder. Im2Latex further uses finetuning on handwritten dataset to boost performance.
The number of parameters in Im2Latex, Pix2Text, TexTeller and Sumen-base are 243M, 25M, 300M and 350M respectively. While TexTeller achieves the best overall performance of a BLUE score of 0.76, it trains on 7.5 million image-formula pairs, whereas Im2Latex is comparatively smaller, which is trained on around 500,000 image-formula pairs, while also maintaining a BLUE score of 0.67. Im2Latex also has the advantage of being trained on pre-trained models, which makes it more effective to iterate on.

## 3 Methodology

In this section, we describe our methodology. Section 3.1 describes the Data Augmentation process, and section 3.2 describes the different architectures that we used.

### 3.1 Data Preparation and Augmentation

We trained our model on a publicly available dataset of 500,000 image-formula pairs. Due to hardware constraints, we trained our model on 20% of the dataset, with a train-val-test split ratio of 80:10:10. There is more difficultly in sourcing datasets of photos and handwritten equations. We used the methods in Table 1 for generating photo-like images from rendered equations.

Table 1: Current Photo Simulation Filter

| Steps | | |
|---|---|---|
| # | Name | Description |
| 1 | Rotate | Rotate image by $\theta \sim U(-10°, 10°)$ |
| 2 | Skew | Apply an affine transform $\begin{bmatrix}1 & x & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$ where $x \sim U(-0.3, 0.3)$ |
| 3 | Texture | Key out the white background. Place the black equation text over a randomly chosen segment of a paper-like texture. |
| 5 | Contrast (50% chance to occur) | Lower the image contrast by $x \sim U(0.4, 0.6)$ |
| 6 | Brightness (50% chance to occur) | Brighten the image by $x \sim U(0.05, 0.15)$ |

Synthesizing handwritten data is more complex due to the need for diverse samples for each symbol. The MathWriting [1] dataset extracts character locations from DVI files generated from some LaTeX compilers, and replaces them with digitally handwritten versions. Another similar dataset [13] used sample of symbols drawn in pencil, then arranges them as MathWriting did while introducing some random shifts.
We are motivated to attempt a different method of simulating handwritten data for several reasons. Firstly, prior synthesis of handwritten data requires numerous sample for each symbol. This makes it infeasible to synthesize data for less commonly seen symbols, so models are only able to perform well on high school level math. Secondly, this method struggles to create data for dynamic or contextual symbols, such as the brackets denoting a matrix, or a table of data. These change in size depending on context, so simple replacement is not enough and an algorithm would need to be build for each symbol of this type.
Using filters such as edge detection, decays, and small perturbations is the most scalable solution and

can easily be applied to the current data. A simple blur may also be enough to generalize the model better. We hope that introducing variation into the shape of characters and removing certain features such as serifs will approximate handwriting enough to generalize the model.

Table 2: Current Handwritting Simulation Filter

| Steps | | |
|---|---|---|
| # | Name | Description |
| 1 | Scale Image | Scale image by 5x. |
| 2 | Edge Detect | Use Canny edge detection to remove the interiors of symbols. |
| 3 | Dilate | Apply a 3x3 kernel to each filled pixel to grow the symbols. |
| 5 | Scale Image | Scale image back to the orignal size |
| 6 | Perturbate | Apply 2D perlin noise with a scale of 20 and displacement of 5. The vector for each pixel generated by the noise defines the direction that pixel will be moved. |
| 7 | Interpolate | Bilinear interpolation fills any gaps left from applying a perturbation. |
| 8 | Photo | Apply the photo filter as detailed in the previous table. |

## 3.2 Architecture

We follow the general architecture proposed by Im2Latex. This architecture makes use of an encoder model to extract features from the input and a decoder model which uses these tokens to generate corresponding Latex text. In our approach, we explored various encoder and decoder networks, as outlined in Table 2. Notably, we use Swin and SwinV2 transformers for encoding and we experiment with GPT-2 and BART as decoder models.

By integrating these advanced transformer architectures into our encoder and decoder networks, our method effectively captures complex visual patterns, enhancing performance in OCR applications that involve intricate characters and symbols.

Table 3: Total Parameter Counts

| Models | | |
|---|---|---|
| Encoder | Decoder | Parameters |
| Swin Transformer | GPT-2 | $88M + 137M$ |
| Swin Transformer V2 Tiny | BART | $22M + 139M$ |
| Swin Transformer V2 Tiny | DistilGPT | $22M + 80M$ |
| Swin Transformer V2 Tiny | TinyBERT | $22M + 15M$ |

# 4 Results

In this section, we present the results from our experiments using different models. We have chosen the BLEU [12] score as our evaluation metric. BLEU, which stands for Bilingual Evaluation Understudy, is used frequently for machine translation tasks in NLP to evaluate the generated text by the model. It evaluates the overlap of n-grams in the ground truth and the generated text, emphasizing precision across various n-gram lengths (usually up to four words). To prevent inflated scores due to repeated words, BLEU employs a clipping mechanism that counts each n-gram no more times than it appears in the reference. It also applies a brevity penalty to discourage overly short translations that might achieve high precision by omitting necessary words. The final BLEU score is calculated by computing a weighted geometric mean of the n-gram precisions and adjusting it with the brevity penalty.
We used an RX 7900 XT processor to train our models. We used 20 epochs, a batch size of 12, a learning rate of 0.00004, and other hyperparameters matching those used in Img2Latex. Table 3 shows the results of each model with number of iterations, test loss and the BLEU score. We analyze these results below.

Table 4: Results (Unaugmented Data)

| Models | | Results | | |
|---|---|---|---|---|
| Encoder | Decoder | BLEU | Test Loss | Iterations/s |
| Swin Transformer | GPT-2 | 0.6436 | 0.1992 | ~3 |
| Swin Transformer V2 Tiny | BART | 0.6525 | 0.1572 | ~5 |
| Swin Transformer V2 Tiny | DistilGPT | 0.6521 | 0.1640 | ~5 |
| Swin Transformer V2 Tiny | TinyBERT | 0.3506 | 0.1005 | ~8 |

Table 5: Results (Photo Augmented Data)

| Models | | Results | |
|---|---|---|---|
| Encoder | Decoder | BLEU | Test Loss |
| Swin Transformer | GPT-2 | 0.6224 | 0.1389 |
| Swin Transformer V2 Tiny | BART | 0.5931 | 0.1334 |
| Swin Transformer V2 Tiny | DistilGPT | 0.6167 | 0.1415 |
| Swin Transformer V2 Tiny | TinyBERT | 0.3015 | 0.0932 |

Table 6: Results (Handwriting Augmented Data)

| Models | | Results | |
|---|---|---|---|
| Encoder | Decoder | BLEU | Test Loss |
| Swin Transformer | GPT-2 | 0.6123 | 0.1467 |
| Swin Transformer V2 Tiny | BART | 0.6064 | 0.1497 |
| Swin Transformer V2 Tiny | DistilGPT | 0.5972 | 0.1344 |
| Swin Transformer V2 Tiny | TinyBERT | 0.2908 | 0.0900 |

### 4.1 Analysis

Our baseline, the Swin/GPT-2 model based on Img2Latex, achieves a BLEU score of 0.64. Using the newer Swin v2 Transformer as an encoder, we are able to match this result with a significant reduction in parameters, and increase the rate at which predictions are made by roughly 66%.

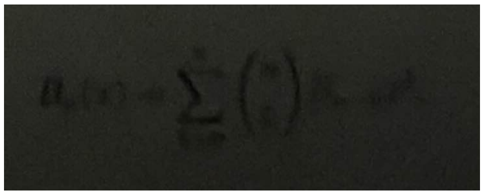
Using more recent decoder models does not appear to have a significant impact on test performance, as replacing GPT-2 with BART did not improve the BLEU score. However, it may be the case that a larger Swin Transformer combined with BART would perform better.

With TinyBERT as the decoder, the BLEU score drops significantly. This is the smallest model we have tried by a significant margin, and may still be useful if deployed on low power devices such as smartphones, especially if trained on a subset of the data with less complexities (i.e. images smaller than 100x100, or labels with $< x$ tokens.).

Lowering the encoder parameter count without lowering the decoder parameter count from SwinV2/BART model would have diminishing returns for overall model size and speed. Therefore, we plan to try some alternative decoders. One candidate is DistilGPT2, with 88M paramaters compared to GPT-2's 137M, and similar general task performance to GPT-2.

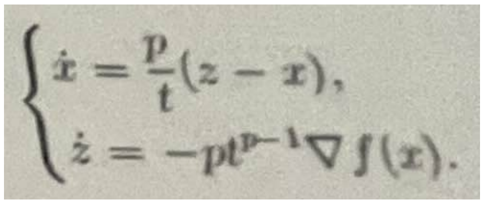
### 4.2 Photo Testing

We tested our most efficient configuration (SwinV2 + DistilGPT) on a small test set of 50 real photos. Many photos are deliberately created in poor conditions, such as low light, harsh angles, and on crumpled paper.



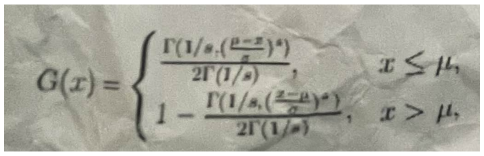
$B_n(x) = \sum_{k=0}^{n} \binom{n}{k} B_k x^{n-k}$

(a) Real Photo w/ poor lighting



$$\begin{cases} \dot{x} = \frac{p}{t}(z - x), \\ \dot{z} = -pt^{p-1}\nabla f(x). \end{cases}$$

(b) Real Photo w/ skew



$$G(x) = \begin{cases} \frac{\Gamma(1/s,(\frac{\mu-x}{\alpha})^s)}{2\Gamma(1/s)}, & x \leq \mu, \\ 1 - \frac{\Gamma(1/s,(\frac{x-\mu}{\alpha})^s)}{2\Gamma(1/s)}, & x > \mu, \end{cases}$$

(c) Real Photo w/ crumpled paper

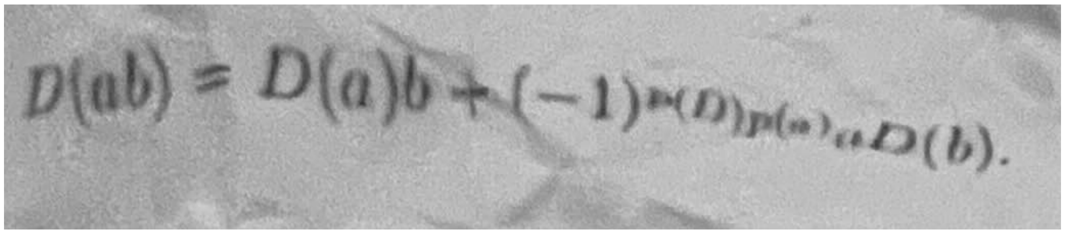
Figure 1: Comparison of real photos under different conditions.

We compare the versions of the model trained on the unaugmented data and the "photo" augmented data. We analyze these results below.

Table 7: Results (Handwriting Augmented Data)

| Training Data | BLEU |
|---|---|
| Unaugmented | 0.3667 |
| "Photo" Augmented | 0.5767 |

The model trained on augmented data performs significantly better, nearly maintaining its performance relative to its training data. This suggests that a photo filter can be a viable substitute for real photos for improving a model's robustness on actual photos.
Below is a sample photo with a result from the unaugmented data model and the augmented data model:



$D(ab) = D(a)b + (-1)^{p(D)p(a)}aD(b).$

Figure 2: Sample Image

**Ground Truth:**

**Render:** $D(ab) = D(a)b + (-1)^{p(D)p(a)}aD(b)$

**Raw:** `D(ab)=D(a)b+(-1)^{p(D)p(a)}aD(b)`

**Unaugmented Data Model Result:**

**Render:** $|D(uv)| \,=\, D^{(1)}(v)\,h + (-1)^{\omega^{s+1}(f)}\,_{f^{(k+1)}}\,_{L^1(V)}\,.$

**Raw:**

```
D(uv)\,=\,D^{(1)}(v)\,h+(-1)^{\omega^{s+1}(f)}\,_{f^{(k+1)}}\,_{L^1(V)}\,.
```

**Augmented Data Model Result:**

**Render:** $D(ab) = D(a)b + (-1)^{m(b)-m(a)}aD(b).$

**Raw:** `D(ab)=D(a)b+(-1)^{m(b)-m(a)}aD(b).`

### 4.3 Handwriting Testing

We tested our most efficient configuration (SwinV2 + DistilGPT) on a small set of partially synthetic handwritten photos. This synthetic data sourced from [13] was created with real samples of handwritten symbols arranged into equations. We compare the versions of the model trained on the unaugmented data, the "photo" augmented data, and the "handwriting" augmented data. We analyze these results below.

Table 8: Results (Handwriting Augmented Data)

| Training Data | BLEU |
|---|---|
| Unaugmented | 0.0511 |
| "Photo" Augmented | 0.0753 |
| "Handwritten" Augmented | 0.0687 |

None of the tested models achieved satisfactory performance on this dataset. Notably, selected outputs did not compile as proper LaTeX source and correlated poorly with the input images.
A likely reason for the poor performance is the differing character shapes. Some characters can be written in ways that perturbations cannot recreate. Testing more varied augmentations, such as changing font for alphanumeric characters, may produce better results.

### 4.4 Pruning

We also experimented with pruning and Mixture of Experts model, which did not yield expected results. Pruning was done on the two of the main models Swin V2 + DistilGPT, and Swin V2 + Bert. Pruning was done according to absolute weight values, and smaller values were pruned. While pruning upto 30% of Swin V2 + DistilGPT and upto 40% of Swin V2 + BERT model reduced the BLEU score by upto 0.05, pruning more than 60% of either model let to a BLEU score decrease of more than 0.1, increasing exponentially after the 60%. However, in our experiments, pruning wasn't able to decrease inference time significantly, neither was it able to reduce memory footprint substantially.

## 5 Conclusion

We developed a robust equation-to-LaTeX conversion system using a hybrid approach that integrates OCR and NLP, leveraging Vision Transformers for feature extraction and decoder language models for LaTeX generation. Our model incorporates preprocessing for noisy inputs, synthetic datasets for handwritten equations, and multi-equation support, achieving robust improvements evaluated with BLEU scores. These advancements make automated mathematical transcription more scalable, efficient, and applicable to real-world conditions. Our results with different encoder and decoder architectures show impressive accuracy, especially when used with photo-augmented data. Future research directions could include leveraging large datasets and designing handcrafted models to enhance performance. Additionally, exploring innovative data transformation techniques presents another promising avenue for further investigation.

## References

[1] Philippe Gervais &Asya Fadeeva &Andrii Maksai (2024) MathWriting: A Dataset For Handwritten Mathematical Expression Recognition. *arXiv*

[2] Nishant Subramani, Alexandre Matton, Malcolm Greaves, and Adrian Lam. A survey of deep learn- ing approaches for ocr and document understanding. arXiv preprint arXiv:2011.13534, 2020.

[3]Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic liter- ature review (slr). IEEE access, 8:142642–142668, 2020.

[4]Piyush Mishra, Pratik Pai, Mihir Patel, and Reena Sonkusare. Extraction of information from handwrit- ing using optical character recognition and neural net- works. In 2020 4th International Conference on Elec- tronics, Communication and Aerospace Technology (ICECA), pages 1328–1333. IEEE, 2020.

[5]R Parthiban, R Ezhilarasi, and D Saravanan. Optical character recognition for english handwritten text using recurrent neural network. In 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), pages 1–5. IEEE, 2020.

[6] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In International Conference on Machine Learning, pages 10347–10357. PMLR, 2021.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10012–10022, 2021.

[9] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo Swin Transformer V2: Scaling Up Capacity and Resolution 2022. arXiv:2111.09883.

[10] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, OBI-TECH Kuo, Colin Raffel, Gerald Shinn, Yejin Tsvetkov, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Processing 2020. arXiv:1910.13461.

[12] Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation." Annual Meeting of the Association for Computational Linguistics 2002.

[13] Tatman, R. (2017). Handwritten mathematical expressions [Dataset]. Kaggle. https://www.kaggle.com/datasets/rtatman/handwritten-mathematical-expressions

[14] Cheema MDA, Shaiq MD, Mirza F, Kamal A, Naeem MA. 2024. Adapting multilingual vision language transformers for low-resource Urdu optical character recognition (OCR). PeerJ Comput. Sci. 10:e1964 DOI 10.7717/peerj-cs.1964

[15] Gowrishankar H, Praveen K S. Optical Character Recognition (OCR): A Comprehensive Review. International Research Journal of Modernization in Engineering, Technology, and Science, 2023

[16] Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF

[17] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu, TinyBERT: Distilling BERT for Natural Language Understanding,*arXiv preprint arXiv:1909.10351*, 2019.

[18] Deng, Y., Kanervisto, A., Ling, J., Rush, A. M. (2016). Image-to-Markup Generation with Coarse-to-Fine Attention. ArXiv. https://arxiv.org/abs/1609.04938

[19] 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

[20] Yang, Z., Zuo, S., Zhou, Y., He, J., Shi, J. (2024). A Review of Document Binarization: Main Techniques, New Challenges, and Trends. Electronics, 13(7), 1394. https://doi.org/10.3390/electronics13071394

[21] Zhang, C., Xu, X., Wu, J., Liu, Z., Zhou, L. (2024). Adversarial Attacks of Vision Tasks in the Past 10 Years: A Survey. arXiv preprint arXiv:2410.23687.

[22] 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