


Article

Investigating Models for the Transcription of Mathematical Formulas in Images

Christian Feichter and Tim Schlippe * 

IU International University of Applied Sciences, 99084 Erfurt, Germany

* Correspondence: tim.schlippe@iu.org

Abstract: The automated transcription of mathematical formulas represents a complex challenge that is of great importance for digital processing and comprehensibility of mathematical content. Consequently, our goal was to analyze state-of-the-art approaches for the transcription of printed mathematical formulas on images into spoken English text. We focused on two approaches: (1) The combination of mathematical expression recognition (MER) models and natural language processing (NLP) models to convert formula images first into LaTeX code and then into text, and (2) the direct conversion of formula images into text using vision-language (VL) models. Since no dataset with printed mathematical formulas and corresponding English transcriptions existed, we created a new dataset, *Formula2Text*, for fine-tuning and evaluating our systems. Our best system for (1) combines the MER model *LaTeX-OCR* and the NLP model *BART-Base*, achieving a translation error rate of 36.14% compared with our reference transcriptions. In the task of converting LaTeX code to text, *BART-Base*, *T5-Base*, and *FLAN-T5-Base* even outperformed *ChatGPT*, *GPT-3.5 Turbo*, and *GPT-4*. For (2), the best VL model, *TrOCR*, achieves a translation error rate of 42.09%. This demonstrates that VL models, predominantly employed for classical image captioning tasks, possess significant potential for the transcription of mathematical formulas in images.

Keywords: mathematical formula transcription; formula-to-text; LaTeX-to-text; image captioning; computer vision; natural language processing; vision-language models



Citation: Feichter, C.; Schlippe, T. Investigating Models for the Transcription of Mathematical Formulas in Images. *Appl. Sci.* **2024**, *14*, 1140. <https://doi.org/10.3390/app14031140>

Academic Editors: Yolanda Blanco Fernández and Alberto Gil Solla

Received: 15 December 2023

Revised: 22 January 2024

Accepted: 25 January 2024

Published: 29 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Speech synthesis has advanced significantly in its ability to read textual content accurately [1]. This progress relies on solid transcriptions to convert written text into spoken language. However, a notable limitation arises when it comes to processing images. While text within images can be transcribed using optical character recognition (OCR) technology, the process of converting visual elements, such as mathematical formulas, into spoken words is more difficult than textual content. This is particularly evident in educational materials and scientific documents where formulas are still often expressed as embedded images. A perfect transcription of formulas in images would not only help make coursebooks with mathematical formulas more accessible for blind people, but also help sighted people who want to familiarize themselves with the course content without reading, e.g., during sporting activities or when driving.

However—to the best of our knowledge—there is no work that tackles the problem of transcribing mathematical formulas in images. Figure 1 illustrates the two possible processes for transcribing mathematical formulas. In the first approach, the image of the formula is converted into LaTeX code using mathematical expression recognition (MER) [2–6]. This LaTeX code then serves as input for a natural language processing (NLP) model, which produces the transcription of the formula. In the second approach, the image of the formula is converted into its transcription using a vision-language model (VL), which is able to combine image analysis and text generation [7].

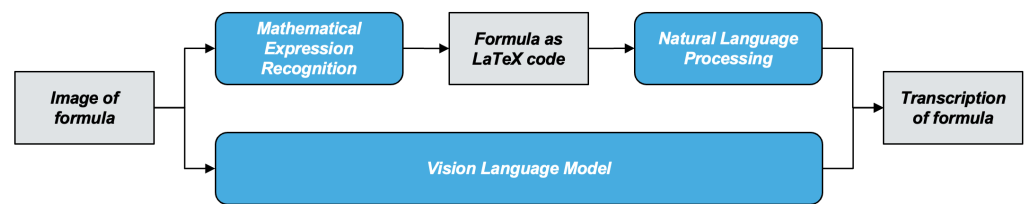


Figure 1. Processes for transcribing mathematical formulas.

Since there was no existing dataset that covered the whole pipeline from the image of the formula as input to the final transcription as output, we created a new dataset called *Formula2Text*, which is publicly available in our GitHub repository (<https://github.com/ficht74/Formula2Text>, accessed on 21 January 2024). *Formula2Text* is based on the existing dataset *IM2LATEX-230K* [8], which we reduced, cleaned, and augmented with English transcriptions.

Consequently, our contributions are as follows:

- We are the first to tackle the problem of transcribing images of mathematical formulas.
- We created a benchmark dataset for this task: *Formula2Text*.
- We investigated state-of-the-art MER, NLP, and VL models for this task.
- We share our code and dataset with the research community in our GitHub repository.

In the next section, we will describe the challenges in the process of the transcription of mathematical formulas. We will provide an overview of related work in Section 3. In Section 4, we will present our experimental setup. Our experiments and results will be described in Section 5. In Sections 6 and 7, we will conclude our work and indicate possible future steps.

2. Challenges in Reading and Transcribing Mathematical Formulas

Ref. [9] explains that the challenges in reading mathematical formulas lie fundamentally in the use of complex structures, symbols, numbers, relational signs, and different notations.

Figure 2 illustrates the five main challenges a human has to deal with when reading a mathematical formula. These are the same challenges our machine learning models have to deal with in order to retrieve the transcription of the formula. The challenges are as follows:

1. Retrieve the principal symbols (e.g., σ , $=$, $\sqrt{\quad}$, $-$, Σ), even those that are not displayed (e.g., multiplier \cdot).
2. For each principal symbol, find the corresponding sub-symbols that deliver additional information (e.g., $\frac{1}{n}$, $\sum_{i=1}^n$, $(x_i - \mu)^2$).
3. A symbol may have several semantics in different positions, resulting in various possible transcriptions. For each symbol and its corresponding sub-symbol, find the correct transcription (e.g., “one divided by n ” vs. “one over n ”, “sum of” vs. “sum from”).
4. Find the correct order for the individual transcriptions (e.g., “the sum from i is one to n of x_i minus μ ” vs. “the sum of x_i minus μ from i is one to n ”).
5. Retrieve a complete transcription that covers the whole formula, “sigma is equal to the square root of one divided by n times the sum from i equals one to n of x_i minus μ squared”).

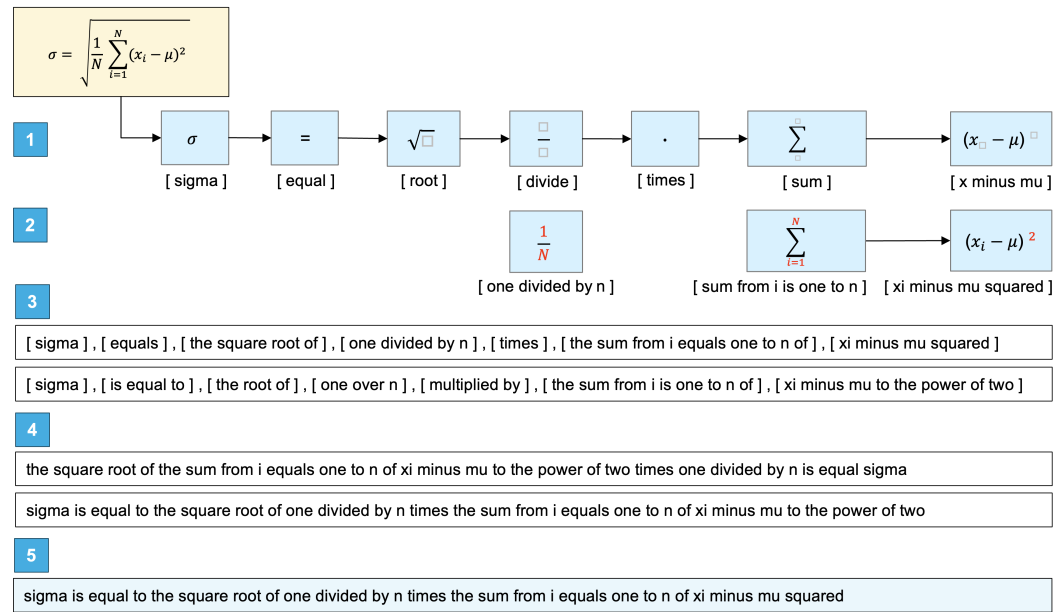


Figure 2. Challenges in reading and transcribing a mathematical formula.

In the automatic transcription of mathematical formulas, *challenge 1* and *challenge 2* are usually handled using MER models, which convert the image of the formula into LaTeX code [2–6]. As demonstrated in Figure 3, LaTeX code usually contains commands such as $\backslash lim$, $\backslash sum$, and $\backslash frac$. These commands still have to be converted into natural language. NLP models have the potential to perform this conversion.

```
\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^N (x_i - \mu)^2}
```

Figure 3. Example of a mathematical formula in LaTeX code.

Alternatively, VL models may be able to deal with *challenge 1–5* as they connect vision and language in a generative way. Therefore, these models support tasks such as image captioning, which translates an input image into a textual description [10,11]. Consequently, we investigate VL models’ potential for the transcription of mathematical formulas in images.

3. Related Work

In this section, we will first describe approaches to convert images of formulas into LaTeX code with the help of MER. Then, we will present approaches of related work that has the potential to transcribe LaTeX code automatically. Finally, we will present how other researchers leverage VL models for different tasks.

3.1. Mathematical Expression Recognition

OCR, i.e., the transcription of normal printed texts in documents and videos, has been extensively studied, e.g., [12,13]. However, conventional OCR systems are not able to process mathematical expressions in documents or images due to the formulas’ special structure, symbols and the position of the formulas’ elements [14]. Therefore, as demonstrated in Figure 4, a specialized mathematical OCR system—called mathematical expression recognition (MER) in literature—is required to convert mathematical content into markup languages such as LaTeX [14].

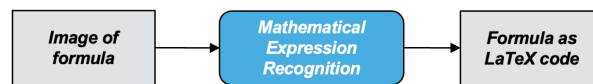


Figure 4. Mathematical expression recognition for LaTeX generation.

In recent years, these systems have evolved rapidly through deep learning. Usually, transformer-based approaches [2–4] have proven to outperform traditional statistical models [15,16] and convolutional neural networks [5,6,17–19]. These neural networks are able to learn and recognize intricate patterns and features within images automatically, making them particularly well-suited for accurately extracting text with subscripts such as mathematical formulas from scanned documents or images [20].

3.2. Natural Language Processing Models

To the best of our knowledge, there is no publication that specifically addresses the creation of transcriptions from LaTeX code. Consequently, assuming that state-of-the-art NLP models can create a transcription from LaTeX code, as shown in Figure 5, we particularly explored approaches that tackle similar tasks, such as generative AI and machine translation.



Figure 5. Transcription of LaTeX code.

According to the website of OpenAI [21,22], *ChatGPT* in version 3.5 is capable of solving natural language math problems, known as *math word problems*, and processing mathematical expressions in LaTeX code. Related models are *GPT-2* [23], *GPT-3.5 Turbo*, and *GPT-4* [24]. Concerning machine translation, as mentioned, there is no related work that deals with LaTeX code. But, for the general task of machine translation, transformer-based models give high performances, e.g., *BART* [25], *T5* [26], and *FLAN-T5* [27]. Ref. [28] presents the *PolyMath Translator*, but this system is only able to translate the words in LaTeX code from one language to another language.

Consequently, we investigated the generative AI models *GPT-2* [23], *ChatGPT* [21] in version 3.5, *GPT-3.5 Turbo*, and *GPT-4* [24] as well as the machine translation models *BART* [25], *T5* [26], and *FLAN-T5* [27] for the task of creating transcriptions from LaTeX code.

3.3. Vision-Language Models

VL models, i.e., the combination of a computer vision model and language model, receive more popularity since they improve pure computer vision tasks and help to deal with tasks that require information from both vision data and language data [29]. As shown in Figure 6, we investigated the potential of VL models for our task of creating transcriptions from formula images.



Figure 6. Generation of transcriptions.

To the best of our knowledge, there is no publication that describes the direct transcription of formulas in images with VL models. However, VL models are used to analyze images and generate text that describes the images [30–32]. This specific task is referred to as *image captioning*. There are also vision encoder–decoder models like *SWIN-GPT-2* (https://huggingface.co/docs/transformers/main/en/model_doc/vision-encoder-decoder, accessed on 21 January 2024), which consists of a vision transformer *SWIN* [33] as encoder and *GPT-2* [23] as decoder, but, in contrast to the VL models mentioned before, both

parts work independently of each other. VL models that demonstrate the highest performances in image captioning are *BLIP* [34] and *GIT* [35]. *TrOCR* [36] outperforms the best state-of-the-art models in recognizing printed and handwritten text and supports image captioning too. Consequently, we analyzed the VL models *BLIP* [34], *GIT* [35], *TrOCR* [36], and *SWIN-GTP-2* [33,37] for the task of creating transcriptions from LaTeX code.

4. Experimental Setup

In this section, we will first describe our dataset *Formula2Text*. Then, we will present the MER, NLP, and VL models that we investigated for the transcription of mathematical formulas in images.

4.1. Our *Formula2Text* Dataset

As there was no pre-existing dataset encompassing the entire pipeline from the input image of the formula to the ultimate transcription output, we developed a new dataset named *Formula2Text*, which is publicly accessible on our GitHub repository. The final *Formula2Text* dataset contains 721 mathematical formula images, corresponding LaTeX codes, and 5 transcriptions for each image, resulting in a total of 3605 transcriptions. Our goal was to build a small, manageable, high-quality dataset. Consequently, we firstly extracted images and corresponding LaTeX codes from the *IM2LATEX-230K* dataset [8], where the LaTeX code contains less than 70 characters, no matrices, no multiple lines, and no systems of equations. Then, we generated the corresponding transcriptions in a semi-automatic way. First, we had *ChatGPT* create initial transcriptions from the LaTeX code. Then, we corrected, cleaned, and normalized the remaining transcriptions, based on our education in higher mathematics. For our experiments, we used 80% of *Formula2Text* for training, 10% for validation, and 10% for testing. Figures 7 and 8 show a *Formula2Text* entry and the corresponding formula.

```
"image_name": "0b59137ceddeacd.png",
"formula": "\\beta = - \\frac { 1 } { 2 \\pi } R",
"transcription1": "beta is equal to the negative one over two pi times r",
"transcription2": "beta is equal to the negative of one over two pi multiplied by r",
"transcription3": "beta is equal to the negative one divided by two pi times r",
"transcription4": "beta is the negative of one over two pi times r",
"transcription5": "beta equals negative one divided by two pi times r"
```

Figure 7. *Formula2Text* entry with image name, LaTeX code, and transcriptions.

$$\beta = -\frac{1}{2\pi}R$$

Figure 8. Formula image corresponding to Figure 7.

Figure 7 demonstrates that the *Formula2Text* transcriptions may vary in terms of prepositions (e.g., “negative one over two pi” vs. “negative of one over two pi.”), synonyms (e.g., “times” vs. “multiplied by”), and verb choice (e.g., “equals” vs. “is” vs. “is equal to”). As described in challenge 3 of Figure 2, the word order in transcribing mathematical formulas may vary as well. Comparing the 5 reference transcriptions of each *Formula2Text* entry in the whole dataset also confirms this variability. For example, in our test set, the average TER of the 5 reference transcriptions for each formula is 45.34%, with a variance of 14.03%, indicating strong differences in the valid transcriptions for each formula.

Looking into the LaTeX codes of our dataset also reveals big differences. Figure 9 displays the distribution of the LaTeX code lengths in *Formula2Text* in terms of number of characters. We see that most entries have LaTeX code between 40 and 70 characters, and the average LaTeX code length consists of 53.2 characters.

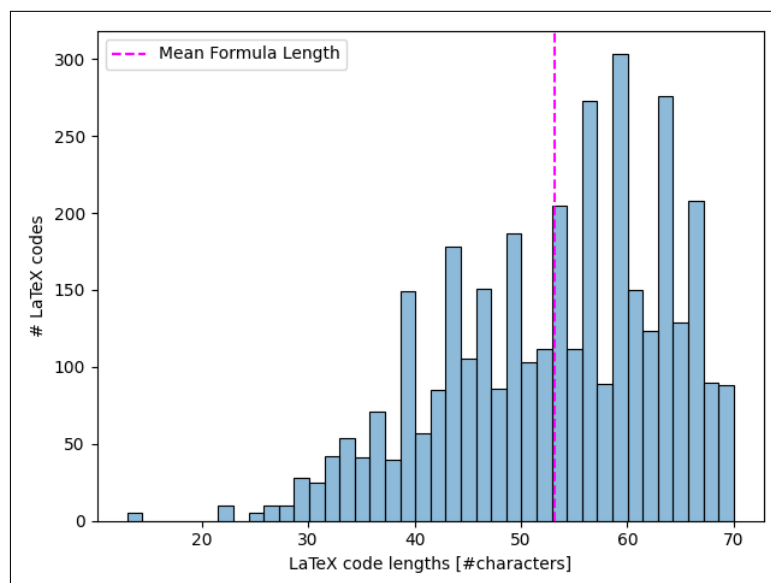


Figure 9. Distribution of the LaTeX code lengths (number of characters).

4.2. Mathematical Expression Recognition Models

To convert formula images into LaTeX code, we analyzed the MER model *LaTeX-OCR* [38], as well as the OCR model *TrOCR* [36], which outperforms other state-of-the-art OCR models [36] and thus has a high potential to be used for MER.

4.2.1. LaTeX-OCR

LaTeX-OCR (<https://github.com/lukas-blecher/LaTeX-OCR>, accessed on 21 January 2024), also known as *pix2tex*, is a transformer-based model that generates LaTeX code from images of mathematical formulas [38]. The model uses a pre-trained vision transformer encoder [39] with a *ResNet* backbone and a transformer decoder [40] trained on the *IM2LATEX-100K* [5] and *CROHME* [41] datasets. [38] reported a BLEU score of 88% on the corresponding test set. An important part of the model is an additional neural network that predicts the optimal resolution for the input image as a pre-processing step.

4.2.2. TrOCR

Compared with *LaTeX-OCR*, *transformer-based optical character recognition (TrOCR)* (<https://huggingface.co/microsoft/trocr-base-printed>, accessed on 21 January 2024) [36] is not an exclusive MER model. *TrOCR* is a pure text recognizer based on the encoder–decoder transformer technology of [40]. The architecture includes both a pre-trained image transformer, *BEiT* [42], for extracting visual features and a text transformer, *robustly optimized BERT pretraining approach (RoBERTa)* [43], for language modeling. This model outperforms the best state-of-the-art models in recognizing printed and handwritten text [36].

4.3. Natural Language Processing Models

As there is no model explicitly developed for the generation of transcriptions from LaTeX code, we investigated NLP models that support the tasks of machine translation or text generation.

4.3.1. T5

Text-to-text transfer transformer (T5) was developed by [26] and has a text-to-text architecture based on the standard encoder–decoder transformer framework from [40]. *T5* allows different NLP tasks such as question answering, machine translation, text classification and summarization, to be handled with the same model, with the same hyperparameters and the same loss function [26]. Therefore, the model requires the name of the NLP task, e.g., “*translate English to German*”, as prefix in addition to the text to process. For our experi-

ments, we trained *T5-Base* (<https://huggingface.co/t5-base>, accessed on 21 January 2024) to execute the new task “*translate LaTeX to Text*”.

4.3.2. FLAN-T5

Fine-tuning Language Net-T5 (FLAN-T5) [27] is a further development of *T5* that leverages more parameters and an enhanced fine-tuning method. The model shows better performance on various benchmark tests compared with previous models [27]. For our experiments, we trained *FLAN-T5-Base* (<https://huggingface.co/google/flan-t5-base>, accessed on 21 January 2024) to execute the new task “*translate LaTeX to Text*” in the same way we trained *T5*.

4.3.3. BART

Bidirectional and Auto-Regressive Transformer (BART) [25] achieves state-of-the-art results in natural language tasks such as text classification, machine translation, text generation, summarizing, and text reasoning. It combines a modified *Bidirectional Encoder Representations from Transformers (BERT)* [44] as the bidirectional encoder and a modified version of *GPT* [23] as the autoregressive decoder. For our experiments, we used *BART-Base* (<https://huggingface.co/facebook/bart-base>, accessed on 21 January 2024).

4.3.4. GPT-2

GPT-2 [23] is a further development of the first *generative pre-trained transformers (GPT)* language model. The model is able to generate new text from the initial text input. We analyzed *GPT-2*, as it can be fine-tuned, which was not possible for newer *GPT* versions at the time of our analyses between mid March and mid August 2023. For our experiments, we used *GPT-2-Medium* (<https://huggingface.co/GPT-2-medium>, accessed on 21 January 2024).

4.3.5. ChatGPT

ChatGPT is also built on the *GPT* language model and was fine-tuned using *reinforcement learning with human feedback*, enabling it to grasp the meaning and intention behind user queries and provide relevant and helpful responses [21]. Although the exact amount of training data for *ChatGPT* has not been published, the previous *GPT-3* model had 175 billion parameters and was trained with 499 billion crawled text tokens, which is substantially larger than other language models [45] like *BERT* [44], *RoBERTa* [43], or *T5* [26]. We experimented with the *ChatGPT* versions *text-davinci-003*, *GPT-3.5-turbo*, and *GPT-4* between mid March and mid August 2023.

4.4. Vision-Language Models

To investigate the complete pipeline from formula images to the transcriptions, we analyzed the VL models *BLIP* [33], *GIT* [35], and *SWIN-GPT-2* [23,33], as they achieve high performances in image captioning and cover different state-of-the-art vision encoders and text decoders. *TrOCR* [36], which we described as the MER model in Section 3.1, can also be categorized as a VL model. Consequently, for *TrOCR*, we analyzed the conversion of formula images to the transcriptions in addition to the conversion of formula images to LaTeX code.

4.4.1. BLIP

Bootstrapping Language-Image Pre-training for unified vision-language understanding and generation (BLIP)'s [34] architecture consists of a multimodal mixture of encoder–decoder. *ViT (Vision Transformer)* [39] is used as the image encoder and *BERT* [44] as the text decoder. *BLIP* supports downstream tasks such as image-text retrieval, image captioning, visual question answering, natural language visual reasoning, visual dialog, and zero-shot transfer to video language tasks. In these tasks, it shows outstanding performance compared with other state-of-the-art approaches [34]. We experimented with *BLIP-Base* (<https://huggingface.co/Salesforce/blip-image-captioning-base>, accessed on 21 January 2024).

4.4.2. GIT

Generative image-to-text transformer (GIT) [35] combines the image encoder *CLIP/ViT-L/14* [46] and the text decoder *BERT* [44] in its *GIT-Large* (<https://huggingface.co/microsoft/git-large-textcaps>, accessed on 21 January 2024) version that we used for our experiments. *GIT* achieves excellent performance in image/video captioning and question answering tasks on various benchmarks, surpassing human performance on the *TextCaps* corpus [47] for the first time [35].

4.4.3. SWIN-GPT-2

Finally, we combined the state-of-art *SWIN Transformer* and *GPT-2* [23] in a VL model leveraging the *Hugging Face Transformer* library (https://huggingface.co/docs/transformers/main/en/model_doc/vision-encoder-decoder, accessed on 21 January 2024). For this, we used the *SWIN-Base Transformer* (<https://huggingface.co/microsoft/swin-base-patch4-window7-224-in22kr>, accessed on 21 January 2024) [33] as a vision encoder to analyze the formula image and generate the image embeddings. As a decoder, we employed *GPT-2* (<https://huggingface.co/GPT-2-medium>, accessed on 21 January 2024) [23] to generate the transcription of the formula image.

4.5. Evaluation Metric

As the *translation error rate (TER)* [48] (also named *translation edit rate* in literature) has higher correlations with human judgments than *BLEU* [49], we used it for our evaluation. The following equation shows the calculation of TER:

$$TER = \frac{\text{\# of edits}}{\text{average \# of reference words}} \quad (1)$$

TER is characterized as the smallest number of edits required to transform a hypothesis into an exact match with one of the references [48]. This value is then normalized by the average length of the references. Potential edits encompass the insertion, deletion, and substitution of individual words, along with rearrangements of word sequences. As suggested by [48], for each hypothesis, we calculate the number of edits for all 5 references and report the optimal (lowest) TER in Section 5.

4.6. Computational Environment for Our Experiments

To perform our experiments as quickly as possible, we ran them in a Google Colab Pro environment (<https://colab.research.google.com>, accessed on 21 January 2024) with Expanded Random Access Memory and A100 Nvidia GPU enabled.

5. Results

In this section, we will present and illustrate the results of the experiments on our *Formula2Text* test set. The test set consists of 80 formula images with associated LaTeX code and five English reference transcriptions. The figures in Section 5 show the average TER (<https://huggingface.co/spaces/evaluate-metric/ter>, accessed on 21 January 2024) score for the entire test set. This means that, in the following figures, the better systems are represented by lower numbers.

Many pre-trained models are already good at several tasks. Consequently, in addition to evaluating the performance of the models fine-tuned on our validation set, our goal was to analyze the performance when no validation set was available for fine-tuning. Therefore, in the figures and tables of this section, we will present the TERs of the pre-trained models without fine-tuning, in addition to the TERs of the fine-tuned models, and report the impact of fine-tuning.

5.1. Mathematical Expression Recognition Models

To convert formula images into LaTeX code, we evaluated the MER models *LaTeX-OCR* [38] and *TrOCR* [36] by comparing their resulting LaTeX codes with the LaTeX codes in the references of the *Formula2Text* test set.

Figure 10 visualizes the performances of the analyzed MER models in terms of TER. The blue bars show the performances of the original pre-trained models. The green bars represent the performances of the models that we fine-tuned using the *Formula2Text* validation set. We observe that the fine-tuned *LaTeX-OCR* achieves, with 23.45%, a notably lower TER compared with the fine-tuned *TrOCR*'s TER of 27.71%, which is 18.16% relative.

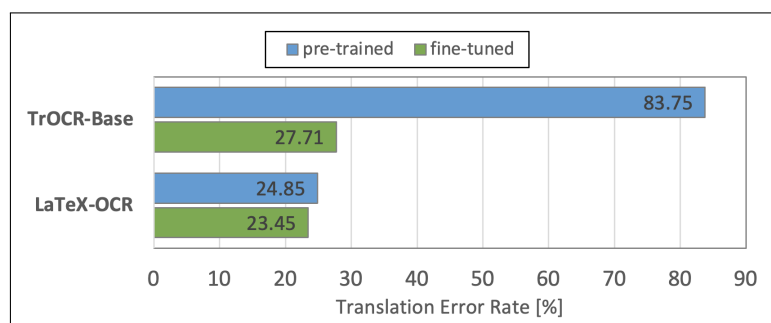


Figure 10. Results of the mathematical expression recognition models.

Table 1 shows the relative improvements with the fine-tuning. We see that the fine-tuning had a significantly higher effect on *TrOCR*, which is explained by the fact that *LaTeX-OCR* had already been trained with printed and handwritten formula images. The fine-tuning process resulted in a substantial relative improvement of 202.24% for *TrOCR* and only in a relative improvement of 5.97% for *LaTeX-OCR*.

Table 1. Impact of fine-tuning the mathematical expression recognition models.

Model	TER [%] Pre-Trained	TER [%] Fine-Tuned	Δ Improvement [%] Relative
TrOCR-Base	83.75	27.71	+202.24
LaTeX-OCR	24.85	23.45	+5.97

5.2. Natural Language Processing Models

To create transcriptions from LaTeX codes, we evaluated seven NLP models plus the online service *MathJAX* (<https://mathjax.github.io/MathJax-demos-web/speech-generator/convert-with-speech.html>, accessed on 21 January 2024) by comparing their resulting transcriptions with the transcriptions in the references of the *Formula2Text* test set. Figure 11 visualizes the performances of the analyzed NLP models in terms of TER. The blue bars display again the performances of the original pre-trained models. The green bars show the performances of the models that we fine-tuned using the *Formula2Text* validation set. Fine-tuning was not possible for *ChatGPT*, *GPT-3.5 Turbo*, *GPT-4*, and *MathJAX* at the time of our experiments from mid March to mid August 2023.

We see that the fine-tuned versions of *T5-Base* (34.95%), *BART-Base* (35.44%), and *FLAN-T5-Base* (35.60%) outperform the other models by far. However, their original pre-trained versions are significantly worse, in the range of 54.44–56.94%. *ChatGPT* achieves 39.95% TER, *GPT-3.5 Turbo* 43.28%, and *GPT-4* 46.01%. Worst performances are obtained by *MathJAX* (56.59%) and *GPT-2-Medium* (76.69). The fine-tuned model of *GPT-2-Medium* has a TER of 63.59%, while its original pre-trained version achieves a TER of 76.69%.

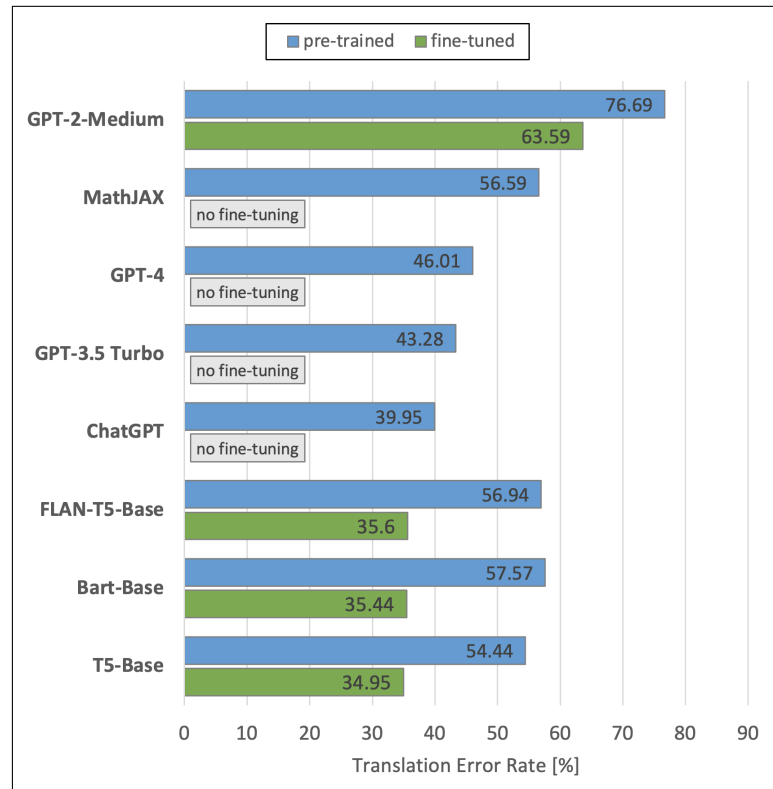


Figure 11. Results of the NLP models.

Table 2 demonstrates the impact of fine-tuning the NLP models. We see that fine-tuning with our validation set helps to achieve relatively lower TERs between 20.60% and even 62.44%. We assume that these differences between the pre-trained and the fine-tuned versions have to do with the fact that the training data of the pre-trained models contained few or no data for the generation of transcriptions from LaTeX code.

Table 2. Impact of fine-tuning the NLP models.

Model	TER [%] Pre-Trained	TER [%] Fine-Tuned	Δ Improvement [%] Relative
T5-Base	54.44	34.95	+55.77
BART-Base	57.57	35.44	+62.44
FLAN-T5-Base	56.94	35.60	+59.94
ChatGPT	39.95	-	-
GPT-3.5 Turbo	43.28	-	-
GPT-4	46.01	-	-
MathJAX	56.59	-	-
GPT-2-Medium	76.69	63.59	+20.60

5.3. Vision-Language Models

To investigate the complete pipeline from formula images to the transcriptions, we analyzed the VL models *BLIP* [34], *GIT* [35], *SWIN-GPT-2* [23,33], and *TrOCR* [36] by comparing their resulting transcriptions with the transcriptions in the references of the *Formula2Text* test set.

Figure 12 displays the TERs of the analyzed VL models. The blue bars display again the performances of the original pre-trained models. The green bars show the performances of the models that we fine-tuned using the *Formula2Text* validation set. We observe that the fine-tuned *TrOCR-Base* performs best with a TER of 42.09%, followed by the fine-tuned *BLIP-Base* (52.18%). Our fine-tuned model *SWIN-GPT* only achieves a TER

of 72.91%. *GIT-Large* shows the worst TERs, with 80.96% after fine-tuning and 82.29% without fine-tuning.



Figure 12. Results of the VL models.

Table 3 gives an overview of the impact of fine-tuning the VL models. We see that fine-tuning with our validation set results in relative improvements of between only 1.64% (*GIT-Large*) and even 104.51% (*TrOCR-Base*). We assume that fine-tuning *GIT-Large* only had such a small impact (+1.64%) since the VL model was pre-trained with datasets that had a focus different from formula images [35] and the amount of samples in our validation set was not sufficient to have a large impact on performance. In contrast, *TrOCR-Base* can already perform the OCR task, i.e., was pre-trained with images containing text. Since the difference between these training data and our formula images is not that big, *TrOCR-Base* can easily learn the task and the impact of fine-tuning is very strong (+104.51%).

Table 3. Impact of fine-tuning the VL models.

Model	TER [%] Pre-Trained	TER [%] Fine-Tuned	Δ Improvement [%] Relative
TrOCR-Base	86.08	42.09	+104.51
BLIP-Base	83.06	52.18	+59.18
SWIN-GPT-2	91.12	72.91	+24.98
GIT-Large	82.29	80.96	+1.64

5.4. Comparing the Combination of MER and NLP Models with Vision-Language Models

Figure 13 presents the TERs of the combinations of our MER models and our NLP models in comparison with the TERs of our VL models. Our goal was to discover which combination leads to the lowest TER for the complete pipeline of transcribing mathematical formulas in images. In the case of the combination of MER and NLP models, an NLP model receives as input the erroneous output of an MER model. The TER of the MER model *LaTex-OCR* is 23.45% (*Variant 1*). The TER of the MER model *TrOCR-Base* is 27.71% (*Variant 2*). The combinations of *LaTex-OCR* and the investigated NLP models are colored in blue, while the combinations of *TrOCR-Base* and the investigated NLP models are colored in purple. In the case of the VL models, there are no combinations with other models and no intermediate steps that provide erroneous input.

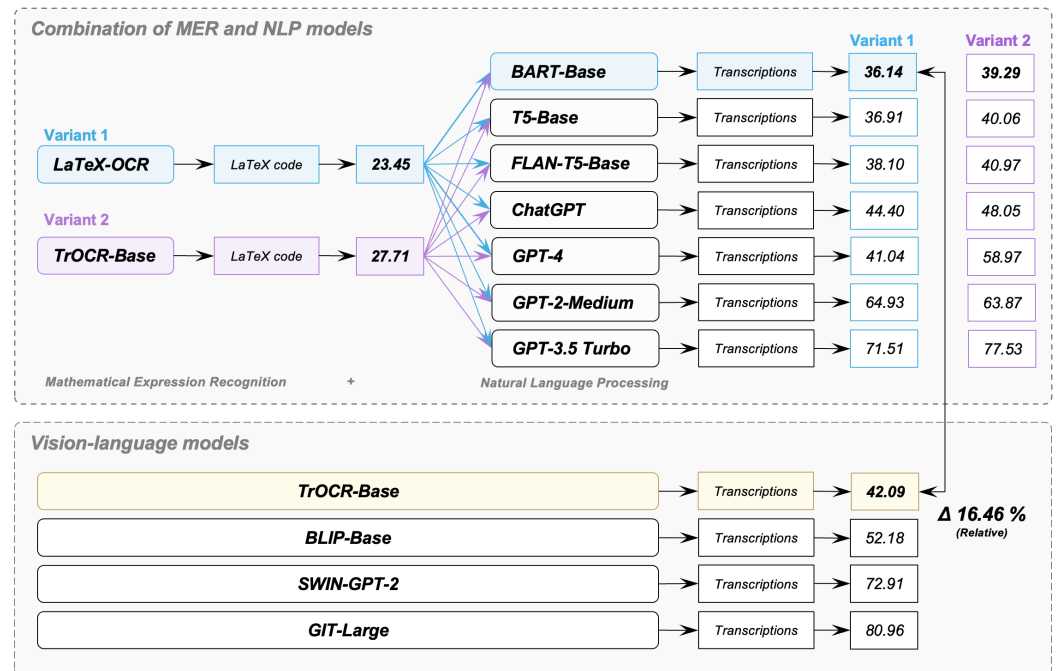


Figure 13. Comparing the combination of MER and NLP models with vision-language models.

We see in Figure 13 that the best six combinations of MER and NLP models achieve TERs that are close to each other (36.14–40.97%). The combination of *LaTeX-OCR* and *BART-Base* results in the lowest TER (36.14%), followed by the combination of *LaTeX-OCR* and *T5-Base* (36.91%). However, our *t*-test demonstrates a non-statistically significant performance difference between the combination of *LaTeX-OCR* and *BART-Base* ($M = 36.14$, $SD = 19.32$) and the combination of *LaTeX-OCR* and *T5-Base* ($M = 36.91$, $SD = 19.45$), where $t(80) = 0.36$ and $p = 0.72 > 0.05$. The third best system is the combination of *LaTeX-OCR* and *FLAN-T5-Base* (38.10%). This is followed by the systems that combine *BART-Base* (39.29%), *T5-Base* (40.06%), and *FLAN-T5-Base* (40.97%) with *TrOCR*. When combining the MER models with the *ChatGPT* versions, the combination of *LaTeX-OCR* and *ChatGPT* version *text-davinci-003* (*ChatGPT*) performs best, with a TER of 44.40%.

The best VL model is *TrOCR-Base*, achieving a TER of 42.09%, which is 16.46% relatively higher than the best system—the combination of *LaTeX-OCR* and *BART-Base*. Our *t*-test demonstrates a statistically significant performance difference between the combination of *LaTeX-OCR* and *BART-Base* ($M = 36.14$, $SD = 19.35$) and *TrOCR-Base* ($M = 42.09$, $SD = 22.00$), where $t(80) = 2.10$ and $p = 0.04 < 0.05$. However, *TrOCR-Base* outperforms seven combinations of MER and NLP models. This demonstrates that a VL model is able to achieve a decent result in the transcription of mathematical formulas in images. The second best VL model is *BLIP-Base*, with a 52.18% TER. *SWIN-GPT-2* achieves a TER of 72.91% and *GIT-Large* of 80.96%.

Comparing the runtimes of the best combination of MER model and NLP model to the runtimes of the best VL model shows that the combination of *LaTeX-OCR* (consisting of 25 M parameters) and *BART-Base* (consisting of 239 M parameters) is on average almost four times faster than *TrOCR-Base* (consisting of 385 M parameters): For transcribing one formula, the combination of *LaTeX-OCR* and *BART-Base* takes only 1.66 s in our Google Colab Pro environment—*LaTeX-OCR* takes 1.15 s and *BART-Base* 0.51 s. In contrast, *TrOCR-Base* needs 6.53 s. This shows that, to use our systems for real-time speech synthesis, the transcriptions have to be generated in advance in order not to create a delay.

5.5. Quality of the Best System's Transcriptions

To give the reader a better idea of the quality of the transcriptions, Table 4 shows the transcriptions for three formulas with different TERs produced by the best system combi-

nation of *LaTeX-OCR* and *BART-Base*. The differences between the system combination's transcription and the closed reference are indicated in bold.

Table 4. Quality of the best system's transcriptions.

Formula Image	Closest Reference	Transcription	TER [%]
$\sigma \rightarrow \frac{\sigma}{\sqrt{\xi}}$	sigma is mapped to sigma divided by the square root of xi	sigma is equal to sigma divided by the square root of xi	8.33%
$ z^2 \ll L^2 \ll \tilde{L}^2$	the absolute value of z squared is much less than l squared, which is much less than tilde l squared	z squared is much less than l squared, which is much more than tilde l squared	25.00%
$\frac{2}{T_H} = \frac{1}{T_L} + \frac{1}{T_R}$	two divided by t h is equal to one divided by t l plus one over t r	two divided by t to the h power is equal to one divided by the quotient of t times l plus one over t times r	44.44%

As shown in Table 4, the quality of the best resulting transcriptions demonstrates that most principal symbols are correctly transcribed (e.g., $-$, $=$, $\sqrt{\quad}$, $+$). However, principal symbols that do not appear so frequently in formulas pose difficulties (e.g., \rightarrow , $|$, \ll). In longer formulas, simple sub-symbols are not always recognized correctly (e.g., T_H , T_L , T_R). However, the order of the words is correct and the structure of the formula is correctly represented.

6. Conclusions

In this paper, we presented our investigation of state-of-the-art models for transcribing printed mathematical formulas from images into spoken English text. This task holds particular relevance for the digital processing and comprehensibility of mathematical content. Our focus encompassed two approaches: (1) The integration of MER models and NLP models to convert formula images initially into LaTeX code and then into text, and (2) the direct conversion of formula images into text using VL models. Since no suitable dataset could be identified that contained both printed mathematical formulas and the corresponding transcriptions, we created the new dataset *Formula2Text* to fine-tune and evaluate our systems. The best performance was achieved using the combination of the MER model *LaTeX-OCR* and the NLP model *BART-Base*. This combination achieved a TER of 36.14%. The best VL model *TrOCR* obtained a TER of 42.09%, resulting in a relative difference of 16.46% compared with the best model combination. Our work shows that decent results can be achieved even with a comparatively small dataset. It also demonstrates that VL models, which are primarily used for common image captioning tasks, have great potential for transcribing mathematical formulas in images—particularly since *TrOCR* was able to outperform seven combinations of MER and NLP models, which represent 50% of the tested systems.

7. Future Work

Our *Formula2Text* dataset could be expanded to include longer, more complex, and handwritten formulas. Moreover, transcriptions in other languages may be included. Due to our limited computational resources, we did not always analyze the model versions with the highest number of parameters in our experiments. Consequently, future work may include the evaluation of further models. But, since we have seen that the runtime for generating transcriptions from formulas on images with deep learning models is relatively long, it should also include applying methods that speed up the execution time of the models, such as pruning, layer fusion, or knowledge distillation. Additionally, it is interesting to combine our systems for the transcription of mathematical formulas in images with speech synthesis and information extraction systems.

Author Contributions: Conceptualization, methodology, software, validation, resources, writing, visualization: C.F. and T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: To contribute to the improvement of transcribing images of mathematical formulas, we share our code and our corpus with the research community: <https://github.com/ficht74/Formula2Text>, accessed on 21 January 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tan, X.; Qin, T.; Soong, F.; Liu, T.Y. A Survey on Neural Speech Synthesis. *arXiv* **2021**, arXiv:2106.15561.
2. Fu, Y.; Liu, T.; Gao, M.; Zhou, A. EDSL: An Encoder-Decoder Architecture with Symbol-Level Features for Printed Mathematical Expression Recognition. In *International Conference on Document Analysis and Recognition*; Springer Nature: Cham, Switzerland, 2020; pp. 134–151.
3. Pang, N.; Yang, C.; Zhu, X.; Li, J.; Yin, X.C. Global Context-Based Network with Transformer for Image2latex. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 10–15 January 2021; pp. 4650–4656. [[CrossRef](#)]
4. Zhou, M.; Cai, M.; Li, G.; Li, M. An End-to-End Formula Recognition Method Integrated Attention Mechanism. *Mathematics* **2023**, *11*, 177. [[CrossRef](#)]
5. Deng, Y.; Kanervisto, A.; Ling, J.; Rush, A.M. Image-to-Markup Generation with Coarse-to-Fine Attention. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 980–989.
6. Wang, J.; Sun, Y.; Wang, S. Image To Latex with DenseNet Encoder and Joint Attention. *Procedia Comput. Sci.* **2019**, *147*, 374–380. [[CrossRef](#)]
7. Uppal, S.; Bhagat, S.; Hazarika, D.; Majumdar, N.; Poria, S.; Zimmermann, R.; Zadeh, A. Multimodal Research in Vision and Language: A Review of Current and Emerging Trends. *arXiv* **2010**, arXiv:2010.09522.
8. Eritsyan, G.; Nawaf, A. im2latex-230k (Version 3) [Data set]. *Zenodo* **2023**. [[CrossRef](#)]
9. Bier, A.; Sroczycioński, Z. Rule Based Intelligent System Verbalizing Mathematical Notation. *Multimed. Tools Appl.* **2019**, *78*, 28089–28110. [[CrossRef](#)]
10. Li, F.; Zhang, H.; Zhang, Y.F.; Liu, S.; Guo, J.; Ni, L.M.; Zhang, P.; Zhang, L. Vision-Language Intelligence: Tasks, Representation Learning, and Large Models. *arXiv* **2022**, arXiv:2203.01922.
11. Du, Y.; Liu, Z.; Li, J.; Zhao, W.X. A Survey of Vision-Language Pre-Trained Models. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI-22*, Vienna, Austria, 23–29 July 2022; pp. 5436–5443. [[CrossRef](#)]
12. Yin, X.C.; Zuo, Z.Y.; Tian, S.; Liu, C.L. Text Detection, Tracking and Recognition in Video: A Comprehensive Survey. *IEEE Trans. Image Process.* **2016**, *25*, 2752–2773. [[CrossRef](#)]
13. Cheng, Z.; Lu, J.; Zou, B.; Qiao, L.; Xu, Y.; Pu, S.; Niu, Y.; Wu, F.; Zhou, S. FREE: A Fast and Robust End-to-End Video Text Spotter. *IEEE Trans. Image Process.* **2021**, *30*, 822–837. [[CrossRef](#)]
14. Aggarwal, R.; Pandey, S.; Tiwari, A.K.; Harit, G. Survey of Mathematical Expression Recognition for Printed and Handwritten Documents. *IETE Tech. Rev.* **2022**, *39*, 1245–1253. [[CrossRef](#)]
15. Suzuki, M.; Tamari, F.; Fukuda, R.; Uchida, S.; Kanahori, T. INFTY: An Integrated OCR System for Mathematical Documents. In *Proceedings of the ACM Symposium on Document Engineering*, New York, NY, USA, 20–22 November 2003; pp. 95–104. [[CrossRef](#)]
16. Zhang, X.; Gao, L.; Yuan, K.; Liu, R.; Jiang, Z.; Tang, Z. A Symbol Dominance Based Formulae Recognition Approach for PDF Documents. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 9–12 November 2017; Volume 1, pp. 1144–1149.
17. Deng, Y.; Kanervisto, A.; Ling, J.; Rush, A.M. What You Get Is What You See: A Visual Markup Decompiler. *arXiv* **2016**, arXiv:1609.04938.
18. Deng, Y.; Yu, Y.; Yao, J.; Sun, C. An Attention Based Image to Latex Markup Decoder. In *Proceedings of the Chinese Automation Congress (CAC)*, Jinan, China, 20–22 October 2017; pp. 7199–7203. [[CrossRef](#)]
19. Yan, Z.; Zhang, X.; Gao, L.; Yuan, K.; Tang, Z. ConvMath: A Convolutional Sequence Network for Mathematical Expression Recognition. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 10–15 January 2021; pp. 4566–4572. [[CrossRef](#)]
20. Wang, Z.; Liu, J.C. Translating Mathematical Formula Images to LaTeX Sequences Using Deep Neural Networks with Sequence-level Training. *Int. J. Doc. Anal. Recognit.* **2021**, *24*, 63–75. [[CrossRef](#)]
21. OpenAI. Introducing ChatGPT. OpenAI, 2022. Available online: <https://openai.com/blog/chatgpt> (accessed on 21 January 2024).

22. Frieder, S.; Pinchetti, L.; Griffiths, R.R.; Salvatori, T.; Lukasiewicz, T.; Petersen, P.C.; Chevalier, A.; Berner, J. Mathematical Capabilities of ChatGPT. *arXiv* **2023**, arXiv:2301.13867.
23. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI blog* **2019**, *1*, 9.
24. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
25. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 7871–7880. [[CrossRef](#)]
26. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1532–4435.
27. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. *arXiv* **2022**, arXiv:2210.11416.
28. Ohri, A.; Schmah, T. Machine Translation of Mathematical Text. *IEEE Access* **2021**, *9*, 38078–38086. [[CrossRef](#)]
29. Gan, Z.; Li, L.; Li, C.; Wang, L.; Liu, Z.; Gao, J. Vision-Language Pre-training: Basics, Recent Advances, and Future Trends. *Found. Trends Comput. Graph. Vis.* **2022**, *14*, 163–352. [[CrossRef](#)]
30. Mokady, R.; Hertz, A.; Bermano, A.H. ClipCap: CLIP Prefix for Image Captioning. *arXiv* **2021**, arXiv:2111.09734. <https://doi.org/10.48550/arXiv.2111.09734>.
31. Lu, J.; Batra, D.; Parikh, D.; Lee, S. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 13–23.
32. Li, L.H.; Yatskar, M.; Yin, D.; Hsieh, C.J.; Chang, K.W. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv* **2019**, arXiv:1908.03557.
33. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, 11–17 October 2021; pp. 9992–10002. [[CrossRef](#)]
34. Li, J.; Li, D.; Xiong, C.; Hoi, S. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 12888–12900.
35. Wang, J.; Yang, Z.; Hu, X.; Li, L.; Lin, K.; Gan, Z.; Liu, Z.; Liu, C.; Wang, L. GIT: A Generative Image-to-text Transformer for Vision and Language. *Trans. Mach. Learn. Res.* **2022**, arXiv:2205.14100.
36. Li, M.; Lv, T.; Chen, J.; Cui, L.; Lu, Y.; Florencio, D.; Zhang, C.; Li, Z.; Wei, F. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *arXiv* **2022**, *37*, 13094–13102. [[CrossRef](#)]
37. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: <https://paperswithcode.com/paper/improving-language-understanding-by> (accessed on 21 January 2024).
38. Blecher, L. LaTeX-OCR: pix2tex: Using a ViT to Convert Images of Equations into LaTeX Code. Available online: <https://github.com/lukas-blecher/LaTeX-OCR> (accessed on 21 January 2024).
39. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30.
41. Mahdavi, M.; Zanibbi, R.; Mouchere, H.; Viard-Gaudin, C.; Garain, U. ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1533–1538. [[CrossRef](#)]
42. Bao, H.; Dong, L.; Piao, S.; Wei, F. BEiT: BERT Pre-Training of Image Transformers. *arXiv* **2021**, arXiv:2106.08254.
43. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2016**, arXiv:1907.11692.
44. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019*; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186. [[CrossRef](#)]
45. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 6–12 December 2020; pp. 1877–1901.
46. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual, 18–24 July 2021; Volume 139, pp. 8748–8763.

47. Sidorov, O.; Hu, R.; Rohrbach, M.; Singh, A. TextCaps: A Dataset for Image Captioning with Reading Comprehension. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
48. Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; Makhoul, J. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, Cambridge, MA, USA, 2006; pp. 223–231.
49. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics-ACL '02, Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.