An AI-powered Mobile App for Mathematics Learning: Enhancing Equation Recognition and Problem-Solving Capabilities

Nam Anh Dang Nguyen[1,2], Binh Nguyen Le Nguyen[1,2], Duy Tan Le[1,2,*], Kha Tu Huynh[1,2], Bao An Mai Hoang[1,2], Hung Nguyen Quoc[3], Phan Hien[3], and Viet Tuyen Nguyen Tan[4]

1. School of Computer Science and Engineering, International University, Ho Chi Minh City, Vietnam
2. Vietnam National University, Ho Chi Minh City, Vietnam
3. School of Business Information Technology, University of Economics Ho Chi Minh City, Ho Chi Minh City, Vietnam
4. School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom

*Corresponding Author: Le Duy Tan (Phone: +84 389-08-18-24; Email: ldtan@hcmiu.edu.vn)

Abstract. We are approaching a future where artificial intelligence (AI) plays a significant role in our society. AI is progressively becoming widespread in many aspects, ranging from industry, manufacturing, energy, and healthcare to education. Among those sectors, education covers significant societal impacts. Mathematics, in particular, presents considerable challenges for both students and individuals who encounter it in their daily professional endeavors. This domain requires learners to have a solid fundamental basis and to be sensitive in integrating knowledge with one another. This study aims to present an AI framework with an application for mathematics problem-solving in education. The designed approach is fine-tuned using a pre-trained model. Indeed, the framework is designed to assist students in their mathematical learning. Our framework can identify individual mathematical characters, formulas, and intricate equations, considering the complex handwritten. By using a hybrid architecture that merges ResNet-v2 and Vision Transformer (ViT) models to improve the accuracy of mathematical character recognition, this model demonstrated its proficiency in rapidly and accurately identifying a wide range of problems, from simple ones to those containing dense and complex mathematical symbols.

Keywords: Optical Character Recognition, Deep learning, Hybrid-ViT, Mathematics

1 Introduction

Optical character recognition (OCR) is a research topic focusing on the automated conversion of text presented in images into text that can be understood

T. A. Trinh et al. (eds.), *Proceedings of the 2nd International Conference - Resilience by Technology and Design (RTD 2024)*, Advances in Intelligent Systems Research 186,
https://doi.org/10.2991/978-94-6463-583-6_3

by machines. According to Gourab Nath [1] in the article "Isolated OCR For Handwritten Forms: An Application in the Education Domain", the applications of OCR in this field can be mentioned as the ability to systematize the learning process [2], especially in mathematics.

Mathematical character recognition technology is more intricate compared to conventional character recognition models. Mathematical expressions typically have a hierarchical structure in which the connections among elements are determined by their placement, size, and specific principles. Therefore, the effective methodology is to generate a latex string corresponding to the input problem image for the final result. However, to make the model understand the rule of latex and keep the original meaning of the expression, it is essential to fine-tune the traditional OCR process.

Mobile applications are programs that run on smartphones and tablets [3]. Most apps can be downloaded and installed on smartphones immediately, making installation easier. Apps may have fewer functionalities than desktop apps due to iOS or Android limitations. However, the app industry has been popular due to its portability, small size, ease of use, and quick response [2]. Therefore, the mobile platform's strength lies in its convenience and simplicity.

Mathematics applications are still facing limitations. Most of them can only support some fields of mathematics within certain constraints. Specifically, it is challenging to solve more advanced, specialized problems. Consequentially, it may cause users to not be fully supported since the applications they download can only solve common types. Finally, most available applications require users to purchase their services to activate all system features, specifically the solution display feature. Besides, new features, such as recognizing mathematical equations, still have many limitations. The reason is due to the use of inappropriate architectures such as CNNs, Template Matching, etc.

In this paper, we aim to optimize the mathematical character recognition model to gain better recognition capabilities in many situations, such as recognizing the user's handwriting, images with complex backgrounds, or images with low resolution. In addition, this paper introduces our designed mathematics mobile application, which integrates the designed AI models to fully support students in learning mathematics.

2 Literature Review

This section reviews previous studies in the context of mathematical equations recognition models. In recent years, there has been significant progress in developing deep learning algorithms and models to improve the accuracy and efficiency of mathematical OCR models. The following section highlights some related studies in this domain.

In [4], the authors implemented a convolutional neural network for OCR tasks. The strength of this model is its ability to recognize many typefaces, such as handwriting and printed ones. The primary dataset used in this model is the MNIST, a familiar dataset in the machine learning and computer vi-

sion domain [4]. This dataset is often used to evaluate and compare machine learning models' performance in handwritten digit recognition. Although the MNIST dataset provides diverse mathematical characters with different classes, this model trained on MINST can only work well on numbers and basic mathematical characters, such as plus, minus, multiplication, and division. One of the main reasons for this limitation is that their designed CNN (Convolutional Neural Network) approach is highly affected by noise or complex backgrounds in the optical image. Besides, the self-attention mechanism is not implemented in their framework. Consequently, CNN can only learn features such as edges, corners, or shapes of characters without understanding the relationship between those features. Therefore, [4] is not effective in recognizing complex equations with square roots, exponents, or integrals, which requires an understanding of the rules of latex strings.

In contrast, the pix2tex model introduced in [5] demonstrated its effectiveness when it can recognize more mathematical symbols well and equations of higher complexity. However, currently, the model cannot recognize images of the user's handwriting. Our designed framework aims to enhance this study by considering this limitation. It should be noticed that the pix2tex model reported in [5] implemented a ViT architecture combined with CNNs, so it can inherit the strengths, as well as overcome the limitations of a pure CNNs model mentioned above. The pix2tex model can be used in the context of basic math operations, and first-order equations, but can also understand more complex equations, such as 4th-order, 5th-order equations, and 1st-order equations. Programs containing radicals, or problems related to calculus. According to the authors, the model was created to help users copy problems, equations, and mathematical formulas from images as quickly and effectively as possible. So, the main data of the model comes from the im2latex-100k dataset, which includes nearly 100,000 images of printed equations. Consequently, the model can recognize a wide range of mathematical equations, with many different complex symbols. However, the performance is not good when executing with the handwritten images. Although the approach introduced by the authors [5] is promising, one of the main weaknesses is that it is not good at recognizing images with complex backgrounds, equations that are not in a straight line, as well as equation images with low pixels. Thus, our paper aims to introduce an enhanced solution to address this challenge.

Similar to pix2tex, CoMER's model has shown remarkable performance when it can recognize complex equations, including handwriting, and this is also the strength that this research wants to achieve [6]. However, the model can only recognize black-and-white images. In other words, input images with complex details or colors will be difficult to detect.

To summarize, there are several limitations in the previous mathematical equation recognition approaches, including difficulties in recognizing handwriting, affected by complex backgrounds and low-resolution images, as well as drawbacks in the variety of mathematical symbols. As a result, our proposed model

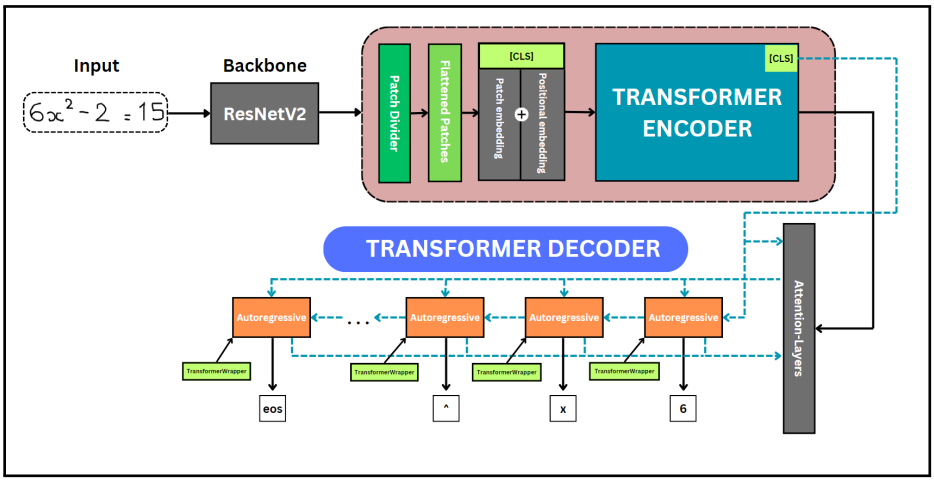

Fig. 1. Hybrid-ViT architecture for mathematics symbol recognition

aims to inherit the advantages while overcoming the limitations of the aforementioned studies.

3 Hybrid-ViT Architecture

Figure 1 illustrates the main architecture of our mathematical character recognition model, namely, Hybrid-ViT. Hybrid Vision Transformer (Hybrid-ViT) combines convolutional neural network (CNN) and Vision Transformer (ViT) components for image processing. The designed model consists of three main parts, including the backbone layer (ResNetV2), the transformer encoder, and the transformer decoder. The following sections will explain the model components in more detail.

3.1 The Backbone Layer – ResNetV2

The ResNetV2 is divided into stem and stages, as illustrated in Figures 2 and 3, respectively. The stem, as presented in Figure 2, is the first part of the network, right after the input layer. The stem is responsible for processing the initial input data and preparing it for the next parts of the network.

In the ResNet Stage, the bottleneck is an important network component, applied to reduce computational complexity and increase model performance. The number of bottleneck layers is based on the number of backbone layers, it includes many main classes, as shown in Figure 3. The backbone layer is configured as LayerBackbone = [2, 3, 7]. Here, the model will have 3 stages, and each stage will correspond to the value in the array. The first stage will have a total of 2 bottleneck layers, resulting in a total of 6 convolutional layers and 6 norm layers. The second stage has 3 bottleneck layers, with 9 convolutional

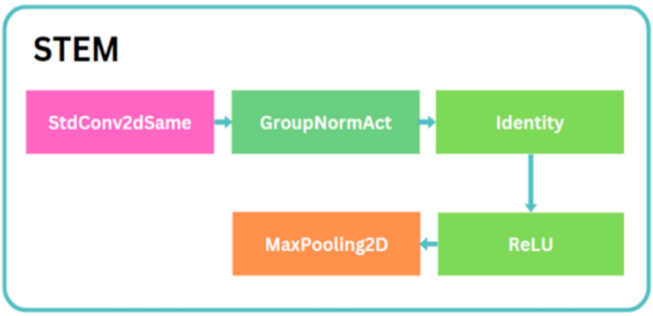

Fig. 2. Stem in ResNetV2

layers and 9 norm layers. The final stage has 7 convolution layers, resulting in a total of 21 convolution layers and 21 norm layers. It should be remarked that a larger number of bottleneck layers allows the model to learn more complex features from the data.

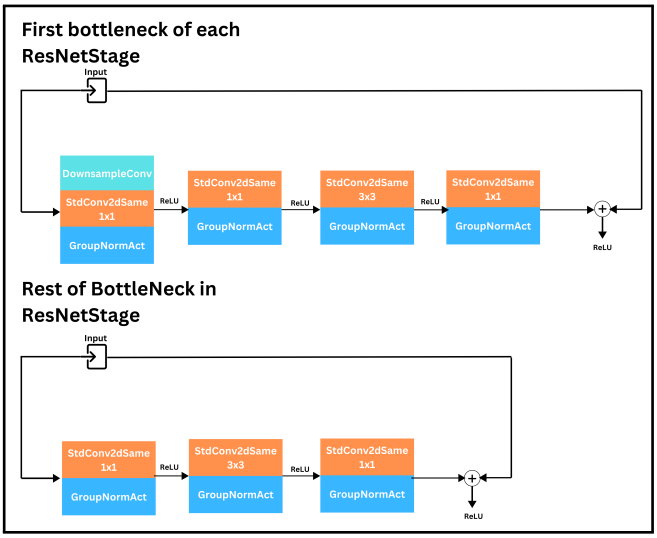

Fig. 3. BottleNeck in ResNetStage

3.2 The Transformer Encoder and Decoder

In the encoder, there is an extra token called x-class, which serves as a learnable embedding. This token, inspired by the concept used in BERT, is referred to

as the [CLS] token. In the training process, the [CLS] token is combined with other spatial tokens in the current patch embeddings. Through a self-attention mechanism, information is compelled to flow from all other tokens to the [CLS] token [7]. Consequently, the embedding of the [CLS] token can be regarded as a comprehensive representation of image features. It can be utilized as an initial hidden state for the model's decoder, instead of using the entire encoder's output feature maps. So, it can be observed that this token represents the first character in the string that will be predicted.

The feature vectors provided by ViT serve as an input to the Transformer decoder. These features contain information about the image that the decoder will use to generate the character string. In the autoregressive process, the decoder generates tokens one by one, each time based on the sequence of previously generated tokens and feature vectors from ViT. The attention mechanism is used to determine the relationship between the generated tokens and the feature vectors [8]. For details:

- **Self-Attention Layer:** This component allows the model to consider every position in the input sequence to produce an output. This mechanism is often implemented with multiple attention "heads" running in parallel, allowing the model to focus on many different relationships at the same time, called Multi-head Attention. It creates a context for each token generation step, helping the model understand the relationships between tokens and generate more accurate predictions.
- **Feed-forward Neural Network:** This linear neural network is implemented after self-attention. It usually consists of two layers with a non-linear activation function in between. The feed-forward layer plays a vital role in processing and refining information after it is passed through the attention mechanism. Each unit in the feed-forward layer processes data from a specific position in the input chain independently of other positions. In other words, this network does not share information between different locations in the chain. This factor supports the model in learning more complex representations, based not only on relationships between tokens but also on the unique characteristics of each token.

After that, a linear layer converts the output vector from the last transformer block into a vector of size equal to the vocabulary size [8]. Then, the softmax layer converts this vector into a probability distribution over all possible tokens in the vocabulary. The model will choose the next token with the highest probability. Here, the model uses top k, with $k = 5$, which will help retrieve five values with the highest probability distribution to make the final prediction for the character [9].

4 Mobile App Development

The main practical implementation of this paper focuses on five major topics in mathematics, including three fundamental fields, which are algebra, calculus, and geometry, and two specialized sectors for numerical methods and probability.

4.1 Client

On the front-end side, the system uses React Native as the main development tool. React Native is a mobile application development framework that helps developers build cross-platform mobile applications [10]. React Native quickly became an essential tool for effective mobile application development. Choosing React Native for client-side development is a suitable choice because this framework has met the set criteria.

4.2 Server

Python matches with software that requires robustness and stability on the server side in data processing [11]. A math application is no exception when its main task focuses on calculating and solving users' problems. The entire project workflow, from receiving requests from users to responding to the final results, is shown by the process diagram in Figure 4.

To create detailed steps for each problem, we applied AST (Abstract Syntax Tree) in Python to extract the components of the equation, such as exponents, square roots, or logarithms. Figure 5 demonstrates a pipeline using the AST tree to extract an expression.

5 Experimental Results and Discussion

5.1 Mobile Application for Mathematics Problem Solving

Our proposed system is currently capable of solving math-related problems using basic mathematics operators, calculus, numerical methods, etc. The image on the left in Figure 6 illustrates the system's keyboard interface, where users will directly enter data to send their math problems to the system for processing. The image on the right is the result returned by the system, including answers and solutions, along with function graphs.

In addition to two advanced fields, numerical methods, and probability, we successfully designed twenty different algorithms to support users, especially university students in related majors. The supported algorithms include Optimization, Ordinary Differential Equations, Curve Fitting methods, etc.

5.2 Datasets

Initially, the pre-trained model pix2tex uses the im2latex-100k dataset, a built-in dataset for the image-2-latex system employment on OpenAI, over 100,000 formulas and pictures are divided into test, validation, and train sets. To enhance the performance, we combined the original pix2tex dataset with CROHME-2014, CROHME-2016, and Linear, Quadric equation datasets. CROHME (Competition on Recognition of Online Handwritten Mathematical Formulas) consists of around 12,000 images from handwritten equations of CROHME. On the other

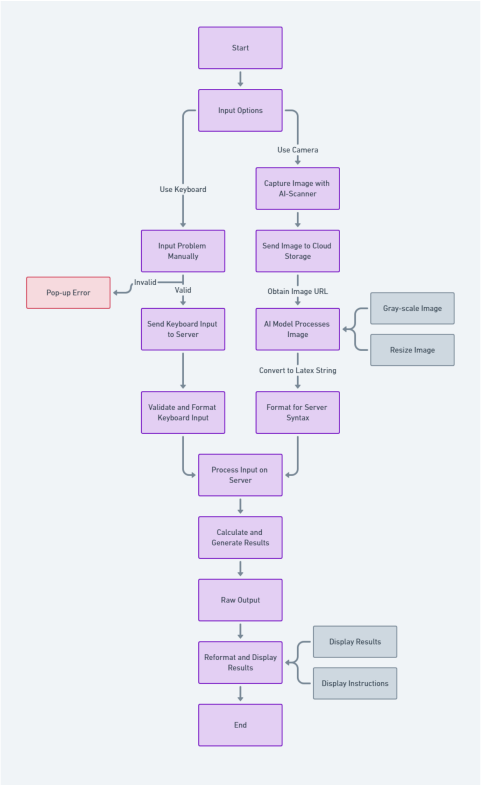

Fig. 4. Process diagram

hand, Linear and Quadratic equation datasets from Kaggle cover 8,000 printed-equation images. The purpose of adding these two datasets is to balance the data for the model. While the majority of the data from image-2-latex and CROHME are mostly random equations, with many less common mathematical symbols, increasing data on simple, common equations helps reduce confusion when the model encounters complex equations, because it has learned basic and common features in simple equations.

5.3 Training the Designed Framework

The training data is stored in the "train.pkl" file and divided into batches of size 10. Each vector embedding has a dimension of 256, and each image patch is sized at 16x16 pixels. The data is encoded into 8000 tokens, including special tokens such as the beginning-of-sequence (BOS), end-of-sequence (EOS), and padding (PAD) tokens. The training parameters were defined based on our empirical experiment. Specifically, the Adam optimization algorithm [12] is employed with a learning rate of 0.001. The loss function is optimized with $\beta_1 = 0.9$ and $\beta_2 =$

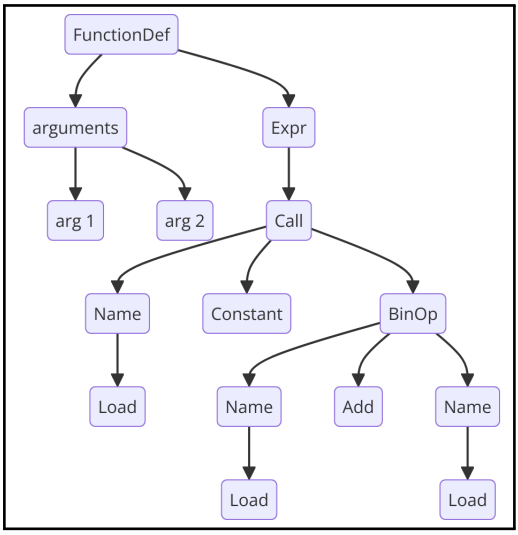

Fig. 5. AST tree

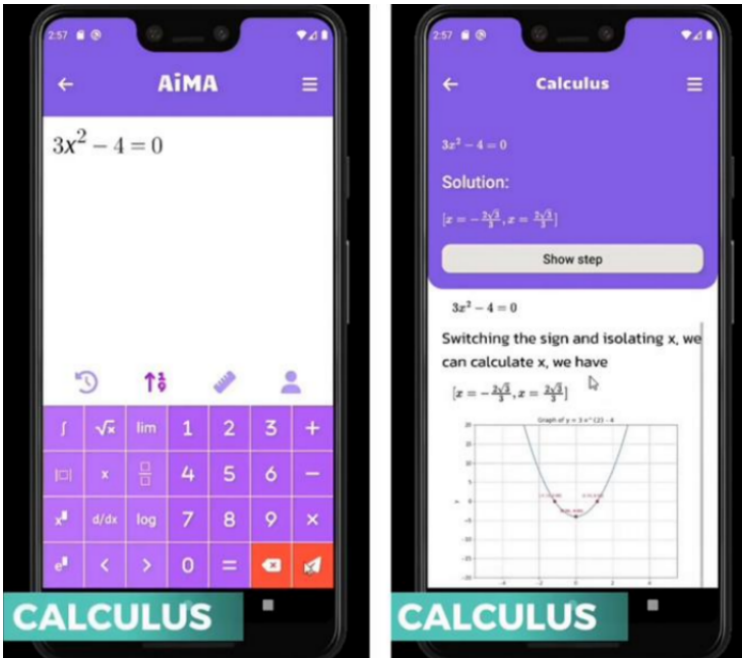

Fig. 6. Calculation and Results screen

0.999. The training process is configured for 50 epochs, with the learning rate adjusted every 30 epochs using the StepLR (Step Learning Rate) algorithm.

5.4 Experimental Results and Discussion

The experimental results are reported in Table 1. The model performance was evaluated using the BLEU score. BLEU (Bilingual Evaluation Understudy) score evaluates the similarity between a predicted sentence and a ground truth by comparing their n-grams. A higher BLEU score indicates the more similar the predicted sentence is to the actual sentence, mapping the system's translation quality. The model's accuracy after training fluctuates from 57% to 58% and the picked checkpoint is about 57.44%, while the BLEU score achieves 89%. In the mathematical equation recognition problem, BLEU score is used to calculate the similarity of the predicted latex string to the label, while the accuracy rate evaluates the percentage of correct predictions compared to the total number of predictions on the test dataset. Therefore, to be able to evaluate the model most accurately, using both is necessary. With this accuracy, it is relatively not high. One of the potential problems is that a mathematical equation can be represented by a latex string in many ways. For example, trigonometric characters, "sin" and "cos", which can also be written as \operatorname{sin}, \operatorname{cos}, instead of \sin, \cos, respectively. In terms of display and meaning, they are the same; however, they are different in terms of raw labels. Consequently, such differences negatively influenced the evaluation process, when comparing the patterns with sample labels. To overcome this limitation, we additionally evaluate the results using the BLEU score, which can better reflect the similarity between two-character strings. The results in Table 1 indicate that our proposed method achieved a percentage of 89% for the BLEU score and about 58% for accuracy in terms of recognizing mathematical equations.

Table 1. Evaluation metrics

Model	Accuracy (%)	BLEU Score (%)	EDA (%)
Pretrained model	57.44	88	90
Our model	58.63	89	90

This is better than when we retrained this model only on the im2latex-100k dataset from scratch, whose accuracy is 57.44% and BLEU score is around 88%. In addition, we make a comparison of the performance of the proposed model with others.

The experimental results indicated that our proposed AI model can recognize a wider range of inputs than other models, including both handwriting images and images with multiple complex backgrounds. Following the retraining process using the new datasets, the model demonstrated significant enhancements in terms of accuracy, BLEU score, and practical performance. By including data, particularly images of handwritten text, the model can acquire a broader range of handwriting styles compared to relying solely on printed formulae. In addition, the datasets containing linear and quadratic equations are used to achieve a balance between complicated and simple character identification. This approach

aids in minimizing confusion between characters that possess similar shapes. However, it is a challenge to recognize handwritten equations with complex symbols, such as integral formulas combined with square roots or fractions. Besides, identifying an equation that is too long can affect the accuracy of the output. Table 2 indicates the comparison.

Table 2. Comparison of Different Models

Criteria	Our	[4]	[5]	[6]	[7]
Handwritten recognition	x		x	x	
Color images recognition	x			x	
Various symbols recognition	x	x	x	x	x
Integrated to (an) application(s)	x			x	
Integrated with solving modules	x				

There are several potential research directions that could be considered in our future work to enhance the model performance. First, instead of using autoregressive, we will experiment with other algorithms, or regression networks, such as Long Short-Term Memory (LSTM). Second, fine-tuning the backbone layer of the hybrid model will also be considered to improve model performance.

6 Conclusion

The study of mathematics is intrinsically intricate, necessitating learners to consistently enhance their skills in order to have a strong and precise foundation that can be effectively used in practical situations. The impact of mathematics on industry sectors such as information technology, electronics, telecommunications, etc., is indisputable. Hence, mathematical knowledge is an important driver of societal advancement. This paper presents a mathematics mobile application that utilizes an AI approach to assist students in solving math-related issues. The experimental results and comparisons indicated that our approach provides significant benefits compared to other options by allowing users to tackle complex calculus problems and receive tailored solutions for certain mathematical subjects. Moreover, to improve the user experience, AI-driven models are used as the optimal measure. Using the pre-trained network, we applied preprocessing methods, model refinement, and conducted data diversification to improve performance, making this model more suitable for a specific mathematics application.

Future work will focus on further enhancing the AI model's ability to recognize and solve more complex mathematical problems, including those involving multiple steps and various mathematical disciplines. Additionally, we plan to extend the application's functionalities to cover a broader range of mathematical topics and integrate more sophisticated algorithms to improve accuracy and efficiency.

References

1. Nath, G., Aggarwal, R., Pant, V., Golla, K., Sekseria, A., Kumar, M. (2022). Isolated OCR For Handwritten Forms: An Application in the Education Domain.
2. Islam, R., Islam, R., Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & Technology*, 10(6), 72–78.
3. Stefańska, M., Wanat, T. (2017). Benefits from using mobile applications by Millennials–a gender and economic status comparative analysis. In *The Proceedings of XVI International Marketing Trends Conference*. Madrid, Paris-Venice: Marketing Trends Association. Retrieved from www.marketing-trends-congress.com/papers
4. Do, K. T. (2020). Handwritten mathematical expressions recognition. Retrieved from https://github.com/rockydtk/Handwritten_mathematical_expressions_recognition
5. Bletcher, L. (2021). LaTeX-OCR. Retrieved from https://github.com/lukas-blecher/LaTeX-OCR
6. Wood, G. (2021). CoMER: Modeling Coverage for Transformer-based Handwritten Mathematical Expression Recognition. Retrieved from https://github.com/Green-Wood/CoMER
7. Le, A. D. (2022). A Hybrid Vision Transformer Approach for Mathematical Expression Recognition. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*.
8. Luu, R. K., Buehler, M. J. (2023). BioinspiredLLM: Conversational Large Language Model for the Mechanics of Biological and Bio-Inspired Materials. *Advanced Science*, 2306724.
9. lucidrains. (2023). x-transformer. Retrieved from https://github.com/lucidrains/x-transformers
10. Danielsson, W. (2016). React Native application development. *Linköpings Universitet*, 10(4), 10.
11. Van Rossum, G., et al. (2007). Python Programming Language. In *USENIX Annual Technical Conference*, Santa Clara, CA, 41(1), 1–36.
12. Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.