AgenTeX: An Agent-Based System for Mathematical Image Parsing and LaTeX Generation

Feolu Kolawole Nick Rui Yogesh Seenichamy

nickrui@stanford.edu nickrui@stanford.edu nickrui@stanford.edu

AgentOps Hackathon, Stanford University

May 2025

Abstract

This paper presents AgenTeX, a novel multi-agent system that automatically translates images containing mathematical content into LaTeX code. By leveraging OpenAI's GPT-4o model, our system implements a specialized agent-based architecture to parse visual mathematical expressions, generate accurate descriptions, and convert them into valid LaTeX code compatible with KaTeX rendering. We demonstrate the system's functionality through both a command-line interface and an interactive web application built with Streamlit. Our approach addresses the challenges of mathematical expression recognition by using large vision-language models instead of traditional optical character recognition techniques. Experimental results show that the system effectively handles a variety of mathematical notations, from simple equations to complex expressions with specialized symbols. This tool has significant implications for accessibility in mathematical content creation, education, and scientific publishing workflows.

1 Introduction

Mathematical notation is a universal language for expressing complex concepts across scientific disciplines. However, digitizing mathematical content remains challenging, particularly when converting visual representations into structured formats like LaTeX, the de facto standard for typesetting mathematics. Traditional approaches to this problem have relied on specialized Optical Character Recognition (OCR) systems that often struggle with the spatial relationships and symbolic complexity inherent in mathematical expressions.

Recent advancements in multimodal large language models (LLMs) have opened new possibilities for visual understanding tasks. These models can process both images and text, demonstrating remarkable capabilities in interpreting visual information and generating appropriate textual descriptions and structured outputs.

In this paper, we introduce AgenTeX, a system that leverages state-of-the-art vision-language models to address the challenge of mathematical image parsing and LaTeX code generation. Our approach uses a multi-agent architecture where specialized agents handle different aspects of the pipeline: from initial image parsing to final LaTeX code generation. By decomposing the problem into these specialized tasks, we achieve a robust system capable of handling diverse mathematical content.

The key contributions of this work include:

1. A novel agent-based architecture for mathematical image parsing and LaTeX generation

2. An implementation that leverages multimodal LLMs (specifically GPT-4o) for high-quality math expression recognition

3. A flexible system supporting multiple input methods (image upload, URL, or direct text)

4. An interactive web application that provides immediate visual feedback through LaTeX rendering

2 Related Work

2.1 Mathematical Expression Recognition

Translating mathematical expressions from images to machine-readable formats has been studied extensively. Traditional approaches typically follow a pipeline of symbol segmentation, recognition, and structural analysis [Chan and Yeung, 2000]. Systems like InftyReader [Suzuki et al., 2003] and SESHAT [Alvaro et al., 2014] have demonstrated reasonable accuracy but struggle with handwritten notation and complex layouts.

Deep learning approaches have improved recognition capabilities, with convolutional neural networks (CNNs) showing promise for symbol recognition [Zhang et al., 2018]. More recently, transformer-based models have been applied to the problem, treating mathematical expression recognition as a sequence-to-sequence task [Zhao et al., 2021].

2.2 Vision-Language Models

Vision-language models represent a paradigm shift in multimodal understanding. Models like CLIP [Radford et al., 2021] established strong foundations for joint image-text representations, while more recent models such as GPT-4 Vision [OpenAI, 2023] and Gemini [Google, 2023] have demonstrated impressive capabilities in understanding and reasoning about visual content.

These models can process images holistically, capturing both the semantic content and spatial relationships without requiring explicit symbol segmentation, making them particularly suitable for mathematical notation recognition.

2.3 Agent-Based Systems

Agent-based architectures have gained traction in AI systems that require coordinated specialized behaviors. LangChain [LangChain, 2023] and similar frameworks have popularized the concept of autonomous agents with specific capabilities working together to solve complex tasks. This approach allows for modular system design, where each agent can be optimized for its particular subtask.

3 Methodology

3.1 System Architecture

AgenTeX employs a multi-agent architecture that breaks down the mathematical image parsing problem into discrete, specialized tasks. Figure 1 illustrates the overall system architecture and information flow.

The system consists of two primary agents:

1. **ImageParserAgent**: Responsible for analyzing images containing mathematical content and generating detailed textual descriptions

2. **LatexGeneratorAgent**: Converts textual descriptions of mathematical expressions into valid LaTeX code

These agents operate in sequence, with the output of the ImageParserAgent serving as input to the LatexGeneratorAgent. This pipeline architecture allows for focused optimization of each stage and enables error isolation and handling.

3.2 Image Parsing

The ImageParserAgent leverages the GPT-4o model's vision capabilities to extract and describe mathematical content from images. Unlike traditional OCR approaches that require explicit symbol segmentation and recognition, GPT-4o processes the entire image holistically, understanding spatial relationships between symbols and interpreting mathematical notation in context.

The agent is instructed to transcribe mathematical expressions exactly as they appear, without solving, simplifying, or modifying them. This preservation of the original expression is crucial for generating accurate LaTeX representations. The agent converts visual content into detailed textual descriptions that capture all relevant mathematical symbols, structures, and relationships.

3.3 LaTeX Generation

The LatexGeneratorAgent takes the textual description produced by the ImageParserAgent and converts it into valid LaTeX code. This agent is specifically instructed to produce KaTeX-compatible LaTeX, ensuring the output will render correctly in the Streamlit application.

The agent's prompt includes comprehensive guidelines for LaTeX generation, covering:

1. Basic mathematical constructs (fractions, roots, superscripts, subscripts)
2. Mathematical functions and operators
3. Greek letters and special symbols
4. Matrix and piecewise function environments
5. Proper spacing and formatting

The agent is trained to output only the minimal valid LaTeX snippet without explanatory text or unnecessary markup, making the output directly usable in rendering contexts.

4 Implementation

4.1 Technical Stack

AgenTeX is implemented using the following technologies:

- **OpenAI API**: For accessing the GPT-4o model
- **Streamlit**: For the web interface
- **Python**: Core programming language
- **Pydantic**: For data validation and model definition
- **AgentOps**: For tracking and analyzing agent performance

The system uses a modular design pattern, separating concerns across different Python modules:

- `models.py`: Defines data structures using Pydantic

- `tools.py`: Implements utility functions for image processing
- `main.py`: Contains the core agent definitions and flow logic
- `app.py`: Implements the Streamlit web interface

4.2 Agent Design

Both agents are implemented using the `Agent` class from the `openai-agents` library, which provides a standardized interface for defining agent behavior, input/output types, and available tools.

The ImageParserAgent is equipped with the `parse_image` tool, which wraps the OpenAI API call to GPT-4o with appropriate vision-specific parameters. The agent's instructions emphasize verbatim transcription without mathematical interpretation.

The LatexGeneratorAgent uses a comprehensive prompt template that provides detailed guidelines for LaTeX generation with specific examples. This approach ensures consistent, high-quality LaTeX output that adheres to KaTeX compatibility requirements.

4.3 Web Interface

The Streamlit-based web interface offers three primary input methods:

1. **Image Upload**: Users can upload images containing mathematical content directly from their devices
2. **Image URL**: Users can provide a URL pointing to an image with mathematical content
3. **Text Input**: Users can enter mathematical expressions or descriptions directly

For each input method, the interface displays:

- The original input (image or text)
- The parsed textual description
- The generated LaTeX code
- A rendered preview of the LaTeX

The interface also provides error handling and feedback mechanisms to guide users through the process.

5 Evaluation

To evaluate the system's performance, we tested it with a variety of mathematical expressions ranging from simple equations to complex expressions with specialized notation. Our evaluation focused on:

1. **Parsing Accuracy**: How accurately the system captures the mathematical content from images
2. **LaTeX Correctness**: Whether the generated LaTeX code correctly represents the intended expression
3. **Rendering Quality**: How well the generated LaTeX renders in the preview

While a comprehensive quantitative evaluation is beyond the scope of this initial paper, qualitative assessment shows that the system performs well on a range of standard mathematical expressions, particularly for printed (rather than handwritten) content with good image quality.

The system demonstrates particular strengths in handling:

- Basic algebraic expressions
- Fractions and square roots
- Summations and integrals
- Simple matrices

Areas for improvement include:

- Complex multi-line expressions
- Specialized mathematical notation (e.g., certain physics symbols)
- Handwritten mathematics with irregular spacing or styling

6 Discussion

6.1 Limitations

The current implementation of Latex_Agents has several limitations:

1. **Model Dependence**: The system relies heavily on GPT-4o's capabilities, inheriting any limitations or biases present in the underlying model.
2. **Processing Speed**: Using GPT-4o for both image parsing and LaTeX generation introduces latency that may be prohibitive for real-time applications or batch processing of numerous images.
3. **Error Propagation**: Errors in the image parsing stage propagate to the LaTeX generation stage, potentially compounding inaccuracies.
4. **Limited Feedback Loop**: The current design lacks a verification mechanism to confirm that the generated LaTeX correctly represents the original mathematical expression.
5. **KaTeX Restrictions**: By targeting KaTeX compatibility specifically, the system may generate LaTeX that lacks advanced features available in full LaTeX environments.

6.2 Future Work

Several promising directions for future work include:

1. **Feedback Mechanisms**: Implementing a verification loop where rendered LaTeX is compared to the original image to detect discrepancies.
2. **Specialized Training**: Fine-tuning models specifically for mathematical notation recognition could improve performance.
3. **Extended Agent Capabilities**: Adding agents for mathematical content classification, difficulty assessment, and step-by-step solution generation as mentioned in the project documentation.

4. **Performance Optimization**: Exploring more efficient model architectures or inference techniques to reduce latency.

5. **Expanded Input Support**: Adding support for PDF documents, screenshots, and handwritten content.

6. **Educational Applications**: Developing features specifically for educational contexts, such as problem generation, solution verification, and learning analytics.

7 Conclusion

AgenTeX demonstrates the potential of applying multimodal LLMs and agent-based architectures to the challenge of mathematical image parsing and LaTeX generation. By leveraging the visual understanding capabilities of GPT-4o and implementing a specialized agent pipeline, we have created a system that effectively translates mathematical expressions from images into valid LaTeX code.

The system's modular design, multiple input methods, and interactive web interface make it accessible to users with varying technical backgrounds. While limitations remain, particularly around processing speed and handling complex notation, the current implementation provides a solid foundation for future enhancements.

This work contributes to the broader goal of making mathematical content more accessible and machine-readable, with potential applications in education, scientific publishing, and accessibility services. As LLMs continue to advance, we anticipate further improvements in the accuracy and capabilities of systems like Latex_Agents.

References

Chan, K. F. and Yeung, D. Y. (2000). Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition*, 3(1):3–15.

Suzuki, M., Tamari, F., Fukuda, R., Uchida, S., and Kanahori, T. (2003). INFTY: an integrated OCR system for mathematical documents. In *Proceedings of the 5th ACM Conference on Electronic Publishing.*

Alvaro, F., Sánchez, J. A., and Benedí, J. M. (2014). Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models. *Pattern Recognition Letters*, 35:58–67.

Zhang, J., Du, J., and Dai, L. (2018). Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In *24th International Conference on Pattern Recognition.*

Zhao, Y., Bogatyy, I., and Iyyer, M. (2021). Im2Latex: A sequence-to-sequence model for converting images of mathematical expressions to LaTeX code. arXiv preprint arXiv:2105.08086.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning.*

OpenAI (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.

Google (2023). Gemini: A Family of Highly Capable Multimodal Models. Google Research Technical Report.

LangChain (2023). LangChain Documentation. `https://langchain.readthedocs.io/`