Human Interaction and Emerging Technologies (IHIET 2024), Vol. 157, 2024, 221–229
https://doi.org/10.54941/ahfe1005481

Exploring the Use of ChatGPT4 API in Approaching Math Word Problems

M. G. R. Harsavardhan[1], Joe Wong[1], Kenneth Y. T. Lim[2], and Swee Ling Leong[3]

[1]Anglo Chinese Junior College, Singapore
[2]National Institute of Education, Singapore
[3]Ministry of Education, Singapore

ABSTRACT

With the evolving educational landscape precipitated by the COVID-19 pandemic, online education has become increasingly prevalent. Much help is needed to provide innovative solutions to address the challenges faced by both students and teachers during this time of crisis. This paper describes an independent research project conducted by a pair of high school students between April 2023 and February 2024, under the mentorship of a senior research scientist at the National Institute of Education in Singapore. The project investigates various methods of Tesseract OCR text recognition, OpenCV image processing, Flask web development and OpenAI's Large Language Models to improve mathematics-solving applications. Our program extracts text using Tesseract OCR, utilising it as input for the GPT-4 API, enabling a conversational presentation of mathematics problems. Users interact by inputting the image address of the math problem that they would like the AI to solve, and GPT-4 provides solutions with detailed step-by-step explanations. OpenCV improves the provided image's quality such as making the text or diagrams more distinct to reduce the possibility of them being misinterpreted. Through rigorous evaluation by testing with different types of maths problems of varying difficulty, our findings underscore the potential for advanced language models in educational tools, offering interactive and intuitive maths problem-solving experiences. There were a few limitations encountered during experimentation, such as challenges with the extraction of non-Latin alphabets and accuracy of the OpenAI's Large Language Modules when solving more complex diagram problems, highlighting the need for further refinement to enhance the system's robustness and adaptability. Future work involves addressing these limitations to broaden the system's applicability for educational purposes and beyond.

Keywords: Education, Mathematics-solving, Tesseract OCR, OpenCV, GPT4-API

INTRODUCTION

Background and Purpose of the Research Area

In the dynamic tech landscape, Python shines as a versatile language for various projects, from simple web apps to complex machine learning with deep-learning algorithms. Our project leverages Python's potential, merging Tesseract OCR, OpenCV, and the OpenAI API for innovative math problem-solving. Designed to assist students, our approach provides quick, step-by-step answers to both word and diagram problems, showcasing the power of Python in creating comprehensive solutions for diverse educational challenges.

Hypothesis of the Research

Our hypothesis centres on the idea that the integration of optical character recognition (Tesseract OCR) and image processing (OpenCV) with advanced natural language processing (OpenAI API, specifically GPT-3.5 and GPT-4 Vision) can significantly enhance the accuracy and efficiency of maths problem-solving. This amalgamation of technologies will provide solutions and detailed step-by-step explanations, revolutionising how users interact with mathematical content.

METHODOLOGY/MATERIALS

Tesseract OCR

Employing Tesseract OCR, we extracted text, focusing on mathematical expressions and laying the foundation for subsequent processing. Tesseract OCR employs algorithms for line and word detection, character identification, and accommodates variations in spacing due to different fonts (Smith, 2007). The versatility of Tesseract OCR, with algorithms like fixed pitch detection and proportional word finding, is crucial for reading user handwriting with varying fonts (NYU, 2024). These features enhance the accuracy of our program, making it resilient to diverse font styles and attributes.

OpenCV Image Processing

Utilising OpenCV, we captured images from the webcam and sent them to the Tesseract OCR to extract text. However, many challenges existed, such as shadows and varying lighting conditions (Computer vision engineer, 2023), as shown in Figure 1. This affects the Tesseract OCR significantly and prevents it from extracting data accurately (Klippa, 2022).

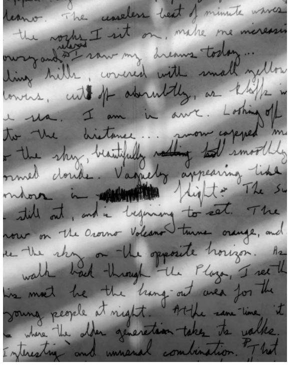


Figure 1: Original image with bad lighting.

To solve this problem, an adaptive threshold is used instead. The algorithm determines the threshold required based on the surrounding pixels (Kumar, 2019). There will be different thresholds calculated for different regions of the image, allowing the text which is darker to become more visible regardless if there is a shadow on the image (Rosebrock, 2021).

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
thresh =
cv2.adaptiveThreshold(gray_image,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY,21,30)
```

This code snippet changes the image to grayscale, then uses adaptive thresholding to make the darker areas more visible, as shown in Figure 2.

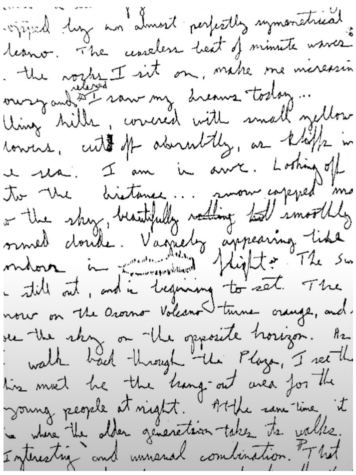


Figure 2: Image after adaptive thresholding.

Through OpenCV's webcam and adaptive thresholding feature, the image can become clearer and users would not need to worry about poor lighting or blocking text when scanning their images.

OpenAI GPT-3.5 API

We integrated OpenAI's GPT-3.5 API to process prompts derived from extracted data. Users could pose mathematical queries, and GPT-3.5 provided detailed, conversational solutions. Our utilisation involved a straightforward API call to GPT-3.5, enabling prompt processing and solution generation within the program. GPT-3.5 is a learning language model that uses transformer neural networks. Neural networks have multiple layers of algorithms where they would recognise the underlying relationships in the input, and the transformer model takes the context into more detail, improving performance on NLP tasks. It analyses natural language queries to predict the best possible response based on their understanding of language. The transformer neural network architecture also uses self-attention mechanisms to focus on different parts of the input text during each processing step. These GPT models rely on the knowledge they gain after they are trained with hundreds of billions of parameters on massive language datasets (AWS, n.d.).

OpenAI GPT-4 Vision API

In addition to GPT-3.5, we explored the capabilities of GPT-4 Vision for image analysis.

```
response = client.chat.completions.create(


  model="gpt-4-vision-preview",
  messages=[
    {
      "role": "user",
      "content": [
        {"type": "text", "text": "What's in this image?"},
        {
          "type": "image_url",
          "image_url": {
            "url":
"https://mindyourdecisions.com/blog/wp-content/uploads/2016/08/harde
st-easy-geometry-thumb-social.png",
          },
        },
      ],
    }
  ],
  max_tokens=300,
)

print(response.choices[0].message.content)
```

This code snippet shows a GPT-4 Vision API call.
This multimodal model integrated image inputs into text, enabling a more comprehensive understanding of the mathematical content presented in images.

Integration of Flask for Interactive Problem Solving

Flask is a simple Python Web framework that has useful tools and features that make website development easier. It provides developers with flexibility and is easily accessible as a website can be easily set up with only one Python file. Flask also offers various customisations with different elements like text boxes and buttons and a wide selection of colours. We leveraged Flask to design an intuitive web interface where users can effortlessly submit math problems for our custom solver. The website features a user-friendly input box where individuals can paste the URL of an image containing a maths problem. This seamless interaction is made possible through Flask's capabilities in handling HTTP requests and rendering dynamic HTML content. Upon submission of the image URL, Flask orchestrates the communication between the user interface and our Python maths problem-solving code. Using Flask routes, we directed form submissions to the appropriate function, initiating the process of solving the maths problem extracted from the provided image.

Dynamic Rendering With Jinja2

Flask's integration with the Jinja2 template engine allowed us to dynamically render HTML templates. The result of the math problem-solving process is seamlessly displayed on the website, providing users with immediate feedback. This dynamic rendering capability enhances the user experience and ensures a fluid interaction with the system.

Scalability and Future Development

The modular structure of Flask ensures scalability and facilitates future enhancements to our system. As our project evolves, Flask provides a solid foundation for incorporating additional features, refining the user interface, and optimizing the overall user experience (Saini, 2023).

In essence, Flask serves as the backbone of our web-based math problem-solving platform, seamlessly connecting users with the power of our custom Python code and OpenAI API integration.

```
from flask import Flask, render_template, request

app = Flask(_name_)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/', methods=['POST'])
def solve_problem():
    url = request.form['url']
    # Call your Python math problem solver function with the provided URL
    # Solve the math problem and get the result
    result = solve_math_problem(url)
    return render_template('index.html', result=result)

if _name_ == '_main_':
    app.run(debug=True)
```

This code snippet shows how Flask is used to set up the website and to call functions when a button is clicked. Figure 3 shows the user interface of the website, in which the user can paste an image URL and submit it in order to get a response.

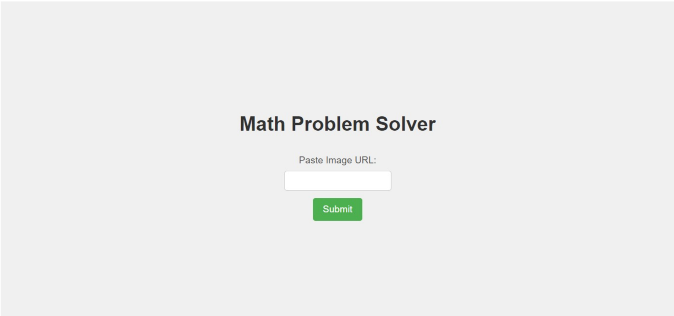


Figure 3: The user interface of the website.

RESULTS/DISCUSSION

We have developed a model using Tesseract OCR to extract text from images, OpenCV to modify these images for the computer to read, and GPT-3.5 to generate answers to the questions from the text extracted. Hence, we investigated the difference between our model and the GPT-4 Vision model.

Test 1 (GPT-4) | Extract Equations From Image | Fail

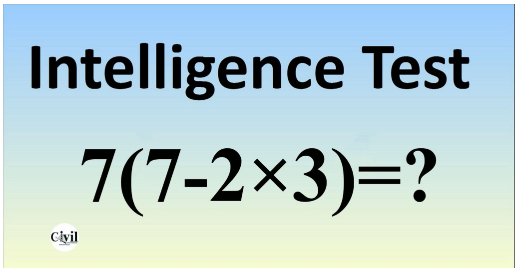
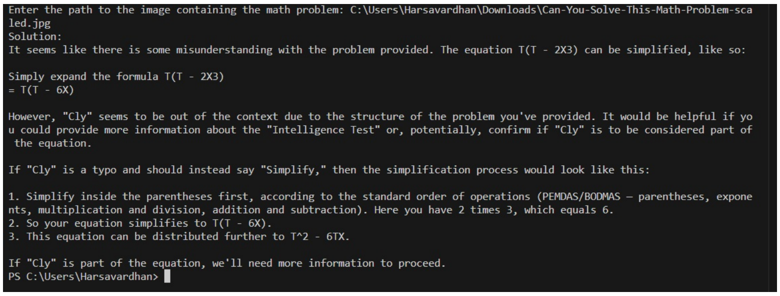


Figure 4: The image (Engineering discoveries, 2020) we tested our model with.

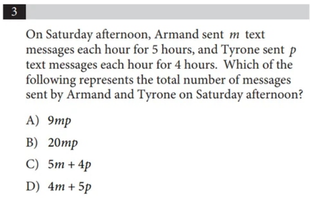


```
Enter the path to the image containing the math problem: C:\Users\Harsavardhan\Downloads\Can-You-Solve-This-Math-Problem-sca
led.jpg
Solution:
It seems like there is some misunderstanding with the problem provided. The equation T(T - 2X3) can be simplified, like so:

Simply expand the formula T(T - 2X3)
= T(T - 6X)

However, "Cly" seems to be out of the context due to the structure of the problem you've provided. It would be helpful if yo
u could provide more information about the "Intelligence Test" or, potentially, confirm if "Cly" is to be considered part of
 the equation.

If "Cly" is a typo and should instead say "Simplify," then the simplification process would look like this:

1. Simplify inside the parentheses first, according to the standard order of operations (PEMDAS/BODMAS – parentheses, expone
nts, multiplication and division, addition and subtraction). Here you have 2 times 3, which equals 6.
2. So your equation simplifies to T(T - 6X).
3. This equation can be distributed further to T^2 - 6TX.

If "Cly" is part of the equation, we'll need more information to proceed.
PS C:\Users\Harsavardhan>
```

Figure 5: The result from our model.

Comments: Heavily inaccurate. Misinterpretation of the equation due to font type. A potential solution involves training a custom machine learning algorithm with various fonts, which was not feasible within the project's time constraints.

Test 2 (Vision) | Text Problem | Success



3

On Saturday afternoon, Armand sent m text messages each hour for 5 hours, and Tyrone sent p text messages each hour for 4 hours. Which of the following represents the total number of messages sent by Armand and Tyrone on Saturday afternoon?

A) $9mp$

B) $20mp$

C) $5m + 4p$

D) $4m + 5p$

Figure 6: The image (Wells, 2019) we tested the vision model with.

```
The image displays a math word problem followed by multiple-choice answers. The problem reads:

On Saturday afternoon, Armand sent m text messages each hour for 5 hours, and Tyrone sent p text messages eac
h hour for 4 hours. Which of the following represents the total number of messages sent by Armand and Tyrone
on Saturday afternoon?

A) 9mp
B) 20mp
C) 5m + 4p
D) 4m + 5p

The correct choice that represents the total number of messages sent by Armand and Tyrone is C) 5m + 4p, sinc
e Armand sent messages for 5 hours and Tyrone for 4 hours.
```

Figure 7: The result from the model.

Comments: GPT-4 Vision demonstrated superior capabilities in solving various problems compared to Tesseract OCR. Notable successes were observed in both text and diagram-based problems.

Test 3 (Vision) | Diagram Problem | Success

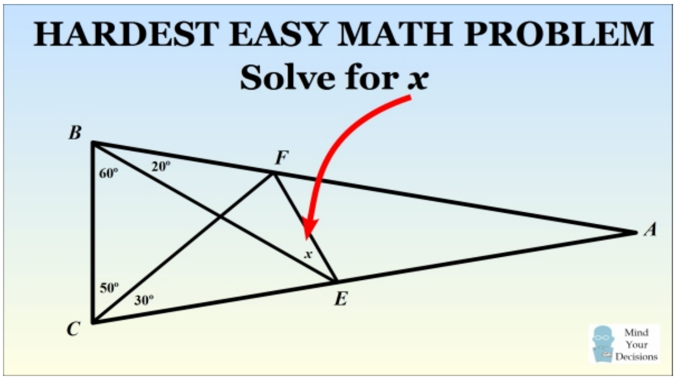


Figure 8: The image (Talwalkar, 2016) we tested the vision model with.

```
The image displays a geometric math problem consisting of a large triangle labeled A, B, C, with smaller tria
ngles inside it. We're asked to solve for the angle x, which is at point F where lines inside the triangle in
tersect. There are various other angles given:

- Angle at point B is 60 degrees.
- There's another angle next to it marked as 20 degrees inside the triangle.
- The angle at point C is 50 degrees, with another angle next to it inside the triangle marked as 30 degrees.

There's a large "HARDEST EASY MATH PROBLEM" caption at the top, and "Solve for x" in the middle with a red ar
row pointing to angle x. It's a puzzle designed to test knowledge of geometry, specifically the properties of
angles within triangles and intersecting lines.

To solve for x, one would typically use the fact that the sum of angles in a triangle is 180 degrees, along w
ith the principles of supplementary and vertically opposite angles. By systematically applying these principl
es and calculating unknown angles step by step, it's possible to determine the value of x. However, without t
he ability to perform calculations or mark on the image, I'm unable to provide the exact solution here.
```

Figure 9: The result from the model.

Comments: GPT-4 Vision was able to identify angles but unable to name angles and solve for x despite knowing the steps to the solution.

CONCLUSION

Future Plans

Unfortunately, due to the limited time constraints, we were unable to implement everything we had hoped to previously. Firstly, we would like to utilise Dall-E, an image generation module developed by OpenAI, to help draw diagrams or models for our users to better understand the working presented by GPT-4. Secondly, we would like to format the large chunk of text provided by GPT-4 into steps that users can easily read and follow. Thirdly, we hope that our app could have been more interactive by asking the user for the next step to follow or to fill in the working after telling the user the next step to solve the question. Lastly, our web application requires some refining in its User interface to be more readable and user-friendly.

Conclusion

In conclusion, our project successfully integrated Tesseract OCR, OpenCV, and the OpenAI API, showcasing the synergy of diverse tools for comprehensive math problem-solving. Tesseract OCR excelled in text extraction, while OpenCV addressed image quality issues, and GPT-3.5 and GPT-4 Vision provided insightful solutions for both text- and diagram-based problems. Our findings underscore the potential of these technologies in creating interactive learning experiences.

However, our experimentation revealed certain limitations. GPT-4 Vision faces challenges with non-Latin alphabets, impacting global optimization. Enlarging small text improves readability but risks cropping essential details. Rotation of text or images may lead to misinterpretation, and struggles with diverse visual elements and spatial reasoning tasks were observed. Future work involves refining our approach to enhance robustness and adaptability, aiming to broaden the system's applicability for educational purposes and beyond.

REFERENCES

AWS (n.d.). *What is GPT AI? - Generative Pre-Trained Transformers Explained*. Available at: https://aws.amazon.com/what-is/gpt/#:~:text=More%20specifically%2C%20the%20GPT%20models,on%20their%20understanding%20of%20language (Accessed: 4 January 2024).

Computer vision engineer (2023). *OpenCV tutorial for beginners | FULL COURSE in 3 hours with Python*. Available at: https://www.youtube.com/watch?v=eDIj5LuIL4A (Accessed: 4 January 2024).

Engineering Discoveries (2020). 'A trip back to high school: Can you solve this math problem?', *Engineering Discoveries*, 13 October. Available at: https://engineeringdiscoveries.com/a-trip-back-to-high-school-can-you-solve-this-math-problem/ (Accessed: 4 January 2024).

Klippa (2022). 'Tesseract OCR: What is it, and Why Would You Choose it in 2023?', *Klippa*, 20 October. Available at: https://www.klippa.com/en/blog/information/tesseract-ocr/ (Accessed: 4 January 2024).

Kumar, S. (2019). 'A straightforward introduction to Image Thresholding using python', *Spinor*, 2 October. Available at: https://medium.com/spinor/a-straightforward-introduction-to-image-thresholding-using-python-f1c085f02d5e (Accessed: 4 January 2024).

NYU (2024). *Tesseract OCR Software Tutorial*. Available at: https://guides.nyu.edu/tesseract (Accessed: 4 January 2024).

Rosebrock, A. (2021). 'Adaptive Thresholding with OpenCV (cv2.adaptiveThreshold)', *PyImageSearch*, 12 May. Available at: https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/ (Accessed: 4 January 2024).

Saini, A. (2023). 'An Easy introduction to Flask Framework for Beginners', *Analytics Vidhya*, 23 August. Available at: https://www.analyticsvidhya.com/blog/2021/10/flask-python/ (Accessed: 4 January 2024).

Smith, R. (2007). 'An Overview of the Tesseract OCR Engine', *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. Curitiba, Brazil, 23–26 September. Los Alamitos: IEEE Computer Society, pp. 629–633. doi: 10.1109/ICDAR.2007.4376991.

Talwalkar, P. (2016). 'The Hardest Easy Geometry Problem – Sunday Puzzle – Mind Your Decisions', *Mind Your Decisions*, 4 September. Available at: https://mindyourdecisions.com/blog/2016/09/04/the-hardest-easy-geometry-problem-sunday-puzzle/ (Accessed: 4 January 2024).

Wells, J. (2019). 'The power of plugging numbers in', *Wells Academic Solutions*, 20 September. Available at: https://www.wellsacademics.com/the-power-of-plugging-numbers-in/ (Accessed: 4 January 2024).