Assisting the Expansion, Promotion, and Release of the 28k Images Dataset at ASSISTments

An Interactive Qualifying Project Report
completed in partial fulfillment of the
Bachelor of Computer Science degree
at Worcester Polytechnic Institute

by
Nathan Anderson
Mahir Sowad

In Cooperation With
The ASSISTments Foundation

Date
10 January 2024

Submitted to

Professor Neil Heffernan
Worcester Polytechnic Institute

This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on the web without editorial or peer review.

Abstract

We worked on the MathNet team at The ASSISTments Foundation to complete five projects: an image cropper and personally identifiable information (PII) blocker for the ASSISTments workbench; a study of the effects of different distance metrics and preprocessing effects on the results of k-means clustering of images; a crop-resistant invisible watermark for images; automating the watermarking and sharing of a 28,000 image dataset; and assessing how much information GPT-4 could gather from images of student math work.

Acknowledgements

We would like to thank the following people and organizations:

- The ASSISTments Foundation, for sponsoring our IQP.
- Neil Heffernan, for supervising our project.
- Sami Baral for advising us and providing guidance throughout our project.

Executive Summary

How can a computer tell the difference between a dog, a cat, and a screwdriver? Released in 2009, the ImageNet dataset aims to solve that issue (“About ImageNet”, n.d.). ImageNet contains over 14 million images classifying over 20,000 objects such as dogs, cats, and screwdrivers (“ImageNet”, n.d.).

The MathNet team at ASSISTments has the goal of releasing a dataset similar to ImageNet, but with K-12 math images. When we first joined the MathNet team, they had already prepared one dataset for release: the 28k Images Dataset, which contained 28,000 images of various K-12 homework submissions. Our purpose on the MathNet team was twofold: assist in the expansion and release of the 28k Images Dataset, and promote the 28k Images Dataset by publishing papers. Over the course of our project, we worked on five different sub-projects:

1. Implement an image cropper and personally identifiable information (PII) blocker into the ASSISTments workbench, in order to make expanding the 28k Images Dataset easier
2. Explore different image clustering methods in order to find common wrong answers in the 28k Images dataset, and release our findings in a paper
3. Create a crop-resistant invisible watermarking algorithm to apply to images in the 28k Images dataset, to safely share the dataset with researchers without it being leaked
4. Automate the sharing of the 28k Images Dataset
5. Assess the ability of ChatGPT-4 to gather information from images from the 28k Images Dataset

Image Cropper and PII Blocker

The Image Cropper and PII Blocker was our first project on the MathNet team. Their goal was to minimize the amount of PII on each image submitted by students in the ASSISTments workbench. When a student submitted an image, via either an “Insert Image” button or a “Take Photo” button, they were given an option to crop their image to just the paper (removing most of the background), and their image was sent to an ASSISTments server running the Detectron2 machine learning model, which allows us to black out any body parts such as hands or faces, which are considered PII.

We were given a primitive version of the Image Cropper and PII Blocker, and our job was to bring it up to the code and UI standards of the ASSISTments workbench. We ran into two issues doing this:

1. Converting the code from the Options API to the Composition API, which required a major refactor of the code.
2. Converting the code from JavaScript to TypeScript, which required many small refactor to the code.

Our final implementations were as follows:

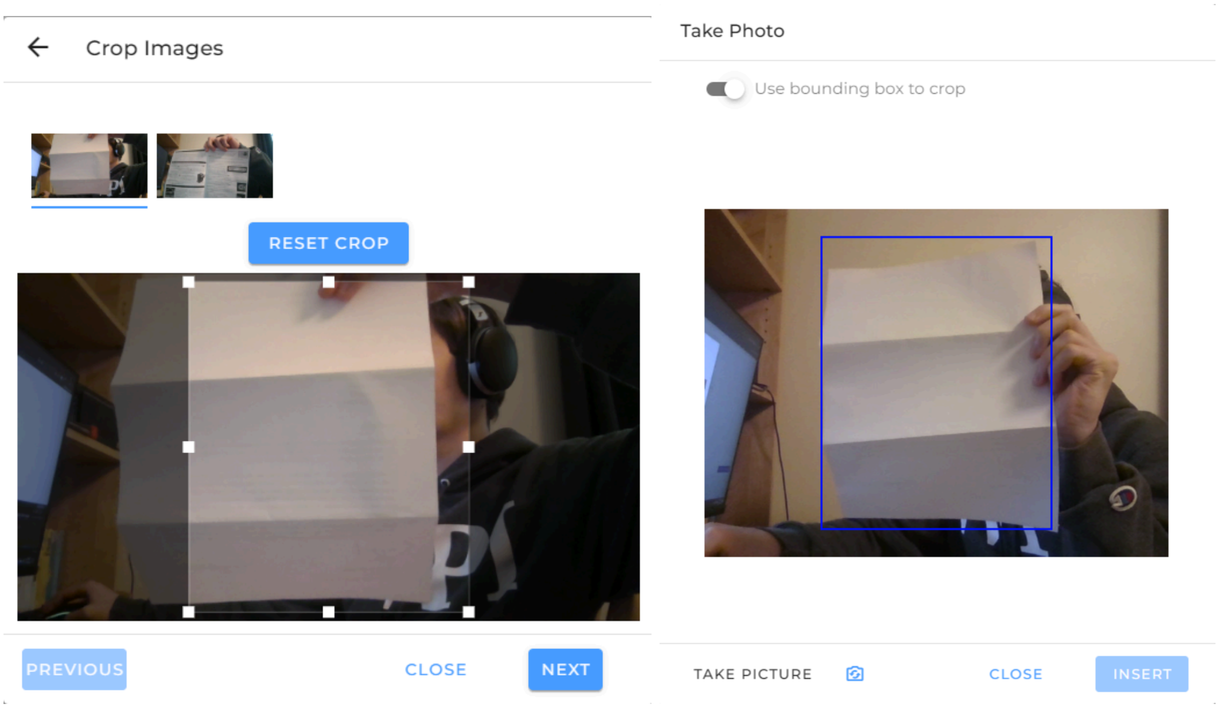


Figure 1: The final implementation of the Insert Image (left) and Take Photo (right) options in StudentWorkbench.

Unfortunately, our final implementation does not have PII blocking functionality since the costs for a server to run Detectron2 were too high.

Image Clustering

We were tasked with using the 28k Images Dataset to find common wrong answers – multiple students submitting the same incorrect answer to a problem – to inspire other researchers to “beat” our results (Pinto, n.d.).

In our work, we explored the effects of four different distance metrics and four different preprocessing effects on the results of k-means clustering, one of the most popular clustering methods. Distance metrics are an equation that allows us to numerically express how closely two images are related – for example, two very closely related images might have a distance of 0.5 while two very unrelated images might have a score of 10.

In order to measure the performance of distance metrics, we combined two different problems into one large dataset and measured how well each distance metric could separate the dataset back into the original two problems.

The best-performing distance metric was a two-step process:

1. Use the Contrastive Language-Image Pretraining (CLIP) machine learning model released by OpenAI to generate vector embeddings for each image. Each vector embedding is a 512-element vector representing the features of one image.
2. Use the angle between the CLIP-generated vector embeddings as a distance metric.

Additionally, we found that comparing the angle between vector embeddings worked best with the "Cropped" preprocessing effect, where each image was cropped to only its relevant content.

Image Watermarking

The ASSISTments Foundation was not confident that none of the images in the 28k Images Dataset contained PII, and so decided to only release the dataset to select researchers. However, there was still a chance that one of the researchers might leak the dataset, which would potentially release many PII-containing images. Our task was to implement an invisible watermark – a watermark that cannot be detected by the human eye – on images in the dataset, to discourage researchers from leaking it.

We found two existing libraries for applying an invisible watermark to images. After running both libraries through a suite of robustness tests, we discovered that neither of them could decode a watermarked image after it had been cropped. It is very likely that researchers will crop some of the images in the 28k Images Dataset, since many of the images contain a very noisy background – see Figure 2 for an example.

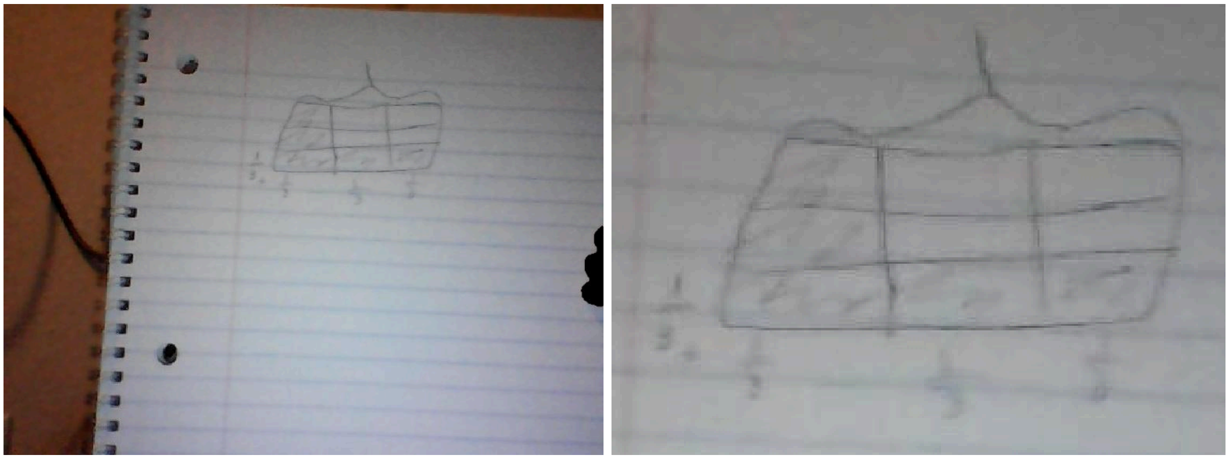


Figure 2: An image with lots of irrelevant background content (left), and the image after it has been cropped to remove the irrelevant content (right).

We modified one of the existing watermarking libraries to support decoding watermarks in cropped images, although it came with two drawbacks:

1. To encode an image, the image must be greater than 128x128 pixels.
2. The message that the images are encoded with must be exactly 32 characters long and must start with '_'.

Dataset Sharing

Watermarking the entire 28k Images Dataset is a tedious process that is better off automated. Each researcher needs their own uniquely-watermarked copy of the dataset, so the entire folder of 28,000 images needs to be copied, which takes hours. After copying is finished, images still need to be downloaded and watermarked manually.

To automate the sharing of the dataset, we chose to use Google Apps Script, which allows for automation within the Google suite (Docs, Drive, Sheets, etc.). Requests to access the dataset are sent via a Google Form. Below is an example of the request process:

1. The researcher requesting the dataset submits the request form.
2. Emails are sent out to the researcher, who is notified that their request has been received, and to a human verifier at ASSISTments, who looks at the submitted form to verify that the request is legitimate.
3. If the human verifier approves the request, a Python server is notified. This server copies and encodes the dataset, then compresses it to a ZIP and uploads it to Google Drive.
4. When the zipped dataset has been fully uploaded, an email is sent to the researcher containing the link to the uploaded ZIP. Another email is sent to the verifier to notify them that the dataset has been shared successfully.

GPT Prompting

With the introduction of the Image Analysis tool, GPT-4 became a multimodal language model capable of handling not only textual input but images as well. This opened up the ability for us to test GPT's capabilities in analyzing images of student math work, particularly open-response questions involving diagrams, graphs, and other visuals. From the testing it is not abundantly evident that GPT is capable enough yet to consistently and accurately produce assessments on these images. It is prone to making minor mistakes in reasoning from time to time, becoming distracted by minor details in the image that does not directly pertain to the overarching purpose or learning goal of presenting students with that question, and failing to detect or mention some crucial details present in a student's answer sometimes.

Despite the issues that are still prevalent, GPT's capabilities can still be fairly helpful for certain aspects of assessing student work and providing some very limited feedback. For graphing or plotting problems, it seems that the use of the Advanced Data Analysis tools can be used to produce plots and graphs that may aid in explanations or feedback provided to students. With the plugins mode and more specifically the Wolfram plugin, GPT can be used to verify calculations and solve algebraic questions that come with complementary graphs and number lines to provide reasoning or support the solution. Since these answers, computations, and graphs come from Wolfram Alpha, a larger degree of trust can be placed on their accuracy than if they were to be solely generated by GPT. However, the major downside of this mode is that the image analysis tool is not available in the plugins mode, meaning that the plugins mode does not make use of the multimodal version of GPT-4 but rather just the textual version. This limits or hinders our ability to extract information from the image, have GPT do an initial assessment, and be able to verify and correct any calculations or solutions to certain math problems all in one integrated pipeline. It seems that for the moment being, GPT's capabilities can help supplement the evaluation and feedback process of these student work images in some areas but is not yet advanced enough to be solely relied on a consistent basis.

Authorship

Section	Primary Author
1. Introduction	Nathan Anderson
2. Background	
2.1 ASSISTments	Nathan Anderson
2.2 Personally Identifiable Information	Nathan Anderson
2.3 28k Images Dataset	Nathan Anderson
2.4 Image Cropper and PII Blocker	Mahir Sowad
2.5 Image Watermarking	Nathan Anderson
2.6 K-Means Clustering	Nathan Anderson
2.7 Dataset Sharing	Nathan Anderson
2.8 GPT Prompting (Using GPT-4)	Mahir Sowad
3. Methodology	
3.1 Overview	Nathan Anderson
3.2 Image Cropper and PII Blocker	All
3.3 Image Clustering	Nathan Anderson
3.4 Image Watermarking	Nathan Anderson
3.5 Dataset Sharing	Nathan Anderson
3.6 GPT Prompting	Mahir Sowad
4. Results // Analysis	
4.1 Overview	Nathan Anderson
4.2 Image Cropper and PII Blocker	All
4.3 Image Clustering	Nathan Anderson

4.4 Image Watermarking..Nathan Anderson
4.5 Dataset Sharing..Nathan Anderson
4.6 GPT Prompting..Mahir Sowad

5. Conclusions & Recommendations

5.1 Image Cropper and PII Blocker..All
5.2 Image Clustering..Nathan Anderson
5.3 Image Watermarking..Nathan Anderson
5.4 Dataset Sharing..Nathan Anderson
5.5 GPT Prompting..Mahir Sowad
5.6 Future Work..Nathan Anderson
Appendix A. Image Clustering Paper..All
Appendix B. The Images in the GPT Prompts...Mahir Sowad
Appendix C. The Responses from GPT Prompting.......................................Mahir Sowad

Table of Contents

1. Introduction..1

2. Background..2

    2.1 ASSISTments..2

        2.1.1 MathNet..2

    2.2 Personally Identifiable Information..3

    2.3 28k Images Dataset..3

    2.4 Image Cropper and PII Blocker..3

        2.4.1 HTML..3

        2.4.2 CSS..4

        2.4.3 JavaScript..4

        2.4.4 Node.js..4

        2.4.5 Python..5

        2.4.6 Axios..5

        2.4.7 Flask..6

        2.4.8 YOLOv7..6

        2.4.9 Detectron2..6

        2.4.10 Canvas API..7

        2.4.11 MediaDevices API..7

        2.4.12 Vue..7

            2.4.12.1 Options API..7

            2.4.12 Composition API..8

        2.4.13 StudentWorkbench..8

2.5 Image Watermarking....8

  2.5.1 Invisible Watermarks....8

    2.5.1.1 invisible-watermark....8

    2.5.1.2 blind_watermark....9

  2.5.2 OpenCV....9

2.6 K-Means Clustering....9

  2.6.1 Noise....10

  2.6.2 Distance Metrics....10

    2.6.2.1 Mean Square Error....10

    2.6.2.2 Oriented FAST and Rotated BRIEF....10

    2.6.2.3 Template Match....11

    2.6.2.4 CLIP Embeddings....11

    2.6.2.5 Angle Between Vectors....11

  2.6.3 Edge Images....11

    2.6.3.1 Canny Edge Detection....11

    2.6.3.2 Noise....12

2.7 Dataset Sharing....12

  2.7.1 Google Apps Script....12

2.8 GPT Prompting (Using GPT-4)....12

3. Methodology....14

3.1 Overview....14

3.2 Image Cropper and PII Blocker....14

  3.2.1 Purpose....14

- 3.2.2 Methods....15
- 3.2.3 Insert Image Option....15
- 3.2.4 Take Photo Option....16
- 3.2.5 Flaws and Limitations....17

3.3 Image Clustering....18

- 3.3.1 Purpose....19
- 3.3.2 Clustering Algorithm....19
- 3.3.3 Preprocessing Effects....19
  - 3.3.3.1 Edge Images....19
  - 3.3.3.2 Cropped Images....21
  - 3.3.3.3 Cropped Edge Images....22
- 3.3.4 Distance Metrics....22
  - 3.3.4.1 Mean Squared Error....23
  - 3.3.4.2 Oriented FAST and Rotated BRIEF (ORB)....23
  - 3.3.4.3 Template Match....24
  - 3.3.4.4 CLIP Embeddings....25
- 3.3.5 Dataset Selection....26
- 3.3.6 Performance Metric....27
- 3.3.7 Flaws and Limitations....27
  - 3.3.7.1 MSE....27
  - 3.3.7.2 ORB....28
  - 3.3.7.3 Template Match....28
  - 3.3.7.4 CLIP....28

3.4 Image Watermarking..28

3.4.1 Purpose...29

3.4.2 Methods...29

3.4.3 Robustness Tests...29

3.4.4 The invisible-watermark Algorithm....................................21

3.4.5 Our invisible-watermark Changes......................................32

3.4.6 Decoding a Cropped Image..33

3.4.6.1 Grid Starting Point...33

3.4.6.2 Pixel Starting Point..34

3.4.6.3 Key Character...35

3.4.7 Flaws and Limitations...36

3.5 Dataset Sharing...36

3.5.1 Purpose...37

3.5.2 Process..37

3.6 GPT Prompting..38

3.6.1 Purpose...38

3.6.2 Initial Prompting Approach and Testing...........................39

3.6.3 Examining the Utility of the DALL-E 3 Image Generation Plugin.............43

3.6.4 Examining the Utility of the Wolfram Plugin in GPT Plugins Mode..........43

3.6.5 Examining the Efficacy of Using Markdown In GPT Prompts...................44

3.6.6 Examining the Efficacy of Zero-Shot and One-Shot Prompting.................45

4. Results // Analysis...50

4.1 Overview...50

4.2 Image Cropper and PII Blocker........50
    4.2.1 Insert Image........50
    4.2.2 Take Photo........51
4.3 Image Clustering........52
    4.3.1 MSE........52
    4.3.2 ORB........53
    4.3.3 Template Match........54
    4.3.4 CLIP Embeddings........55
4.4 Image Watermarking........56
    4.4.1 Preliminary Testing........56
    4.4.2 Final Results........57
    4.4.3 Usage of a “Key” Character........58
    4.4.4 Flaws and Limitations........58
4.5 Dataset Sharing........59
4.6 GPT Prompting........59
    4.6.1 Responses to Initial Prompting Approach and Testing........60
    4.6.2 Analysis of Responses to Testing GPT's Ability to Generate Graphs........63
    4.6.3 Analysis of Responses to Testing the Wolfram Plugin........64
    4.6.4 Analysis of Responses to Prompts Utilizing Markdown........66
    4.6.5 Analysis of Responses to Zero-shot and One-shot Prompting........66
5. Conclusions & Recommendations........69
4.6.5 Analysis of Responses to Zero-shot and One-shot Prompting

5.3 Image Watermarking........70
5.4 Dataset Sharing........70
5.5 GPT Prompting........70
5.6 Future Work........71
    5.6.1 Image Clustering........71
    5.6.2 Image Watermarking........71
    5.6.3 GPT Prompting........72
References........73
Appendices........74
Appendix A. Image Clustering Paper........74
Appendix B. The Images in the GPT Prompts........90
    Appendix B1. The Prompt for Problem ID 1754531........90
    Appendix B2. The Prompt for Problem ID 1755170........93
Appendix C. The Responses from GPT Prompting........97
    Appendix C1. Responses to Prompts for Problem ID 1754531........97
    Appendix C2. Responses to Prompts for Problem ID 1755170........102
    Appendix C3. Responses to Prompts for Problem ID 1622991........110
    Appendix C4. Responses to Prompts Testing GPT's Ability to Generate Graphs........112
    Appendix C5. Responses to Prompts Testing the Wolfram Plugin........117
    Appendix C6. Responses to Prompts Utilizing Markdown........125
    Appendix C7. Responses to Prompts Testing Zero-Shot Prompting........126
    Appendix C8. Responses to Prompts Testing One-Shot Prompting........132

List of Figures

Figure 3.3.3.1.1: An unprocessed image (left) and its corresponding edge image (right)...........21

Figure 3.3.3.2.1: An unprocessed image (left) and its corresponding cropped version (right)....21

Figure 3.3.3.3.1: An unprocessed image (left), its corresponding cropped version (middle), and the final cropped edge image (right)...22

Figure 3.3.4.1.1: Two very different images with a MSE of 2531...23

Figure 3.3.4.1.2: Two very similar images with a MSE of 53.25...23

Figure 3.3.4.2.1: ORB comparison between two very different images with a score of 0.2........24

Figure 3.3.4.2.2: ORB comparison between two very similar images with a score of 1.............24

Figure 3.3.4.3.1: Template Match comparison for between two very different images with a score of 0.25..25

Figure 3.3.4.3.2: Template Match comparison for between two very similar images with a score of 0.98..25

Figure 3.3.4.4.1: CLIP comparison for between two very different images with a score of 2.49...26

Figure 3.3.4.4.2: CLIP comparison for between two very similar images with a score of 2.53...26

Figure 3.3.5.1: A diagram showing the paint required to make three batches of light yellow paint...26

Figure 3.4.3.1: An image that will probably be cropped on all sides...30

Figure 3.4.3.2: The same image as Figure 1, cropped on all sides..30

Figure 3.4.4.1: The bit encoding scheme used by invisible-watermark, with an example 4-bit message...32

Figure 3.4.4.2: The bit encoding scheme from Figure 3.4.4.1, after a mild crop removes one pixel from the right side of the image..........32

Figure 3.4.5.1: The bit encoding scheme used by our modified version of invisible-watermark, with an example 4-bit message..........32

Figure 3.4.5.2: The bit encoding scheme from Figure 3.4.5.1, after a severe crop removes a large amount of pixels from the right side of the image..........32

Figure 3.4.6.1.1: A comparison between the grids of an uncropped (left) and cropped (right) image..........33

Figure 3.4.6.2.1: A comparison between the process of decoding an uncropped (left) and a cropped (right) image..........35

Figure 3.6.2.1: A student answer of problem ID 1622991 that contains the problem description..........41

Figure 3.6.2.2: A student answer of problem ID 1622991 that is very unclear, confusing, and has considerable PII interference..........42

Figure 3.6.6.1: The first student answer provided to GPT for zero-shot prompting..........47

Figure 3.6.6.2: The second student answer provided to GPT for zero-shot prompting..........47

Figure 3.6.6.3: The third student answer provided to GPT for zero-shot prompting..........48

Figure 3.6.6.4: The correct answer example image provided to GPT for one-shot prompting....49

Figure 4.2.1.1: The final design of the Insert Image option..........51

Figure 4.2.2.1: The Take Photo option when the user is lining up the paper (left), and after the user has taken the photo (right)..........52

Figure B1.1: The first student work image provided to GPT for Problem ID 1754531..........90

Figure B1.2: The second student work image provided to GPT for Problem ID 1754531..........91

Figure B1.3: The third student work image provided to GPT for Problem ID 1754531.............91

Figure B1.4: The fourth student work image provided to GPT for Problem ID 1754531...........92

Figure B1.5: The fifth student work image provided to GPT for Problem ID 1754531.............92

Figure B2.1: The first student work image provided to GPT for Problem ID 1755170..............93

Figure B2.2: The second student work image provided to GPT for Problem ID 1755170..........93

Figure B2.3: The third student work image provided to GPT for Problem ID 1755170.............94

Figure B2.4: The fourth student work image provided to GPT for Problem ID 1755170...........95

Figure B2.5: The fifth student work image provided to GPT for Problem ID 1755170.............95

Figure B2.6: The sixth student work image provided to GPT for Problem ID 1755170.............96

Figure C4.1.1: GPT's first attempt at generating a cartesian coordinate plane..........................112

Figure C4.2.1: GPT's attempted correction of the first cartesian coordinate plane produced....113

Figure C4.4.1: GPT's first cartesian coordinate plane produced via the Advanced Data Analysis Tool...115

Figure C4.6.1: Cartesian Coordinate Plane With y = 10 + 5x Graphed....................................116

Figure C5.5.1: Number line representation of the solution of 123 + 456.................................121

Figure C5.6.1: Graph of the Solution to the x^2 - 4x + 4 = 0..124

Figure C5.6.2: Number line of the solution to x^2 - 4x + 4 = 0..125

Figure C6.1.1: Solution to graphing problem prompted using markdown................................126

List of Tables

Table 4.3.1.1: Our results using MSE as a distance metric..53

Table 4.3.2.1: Our results using ORB as a distance metric..54

Table 4.3.3.1: Our results using Template Match as a distance metric...54

Table 4.3.4.1: Our results using the angle between CLIP embeddings as a distance metric........55

Table 4.4.1.1: Results from our robustness tests on invisible-watermark and blind_watermark..57

Table 4.4.2.1: The results of our robustness tests on invisible-watermark, blind-watermark, and our own watermarker implementation...57

1. Introduction

How can a computer tell the difference between a dog, a cat, and a screwdriver? Released in 2009, the ImageNet dataset aims to solve that issue (“About ImageNet”, n.d.). ImageNet contains over 14 million images classifying over 20,000 objects such as dogs, cats, and screwdrivers (“ImageNet”, n.d.).

The MathNet team at ASSISTments was created with the goal of releasing a dataset similar to ImageNet, but with K-12 math images. When we joined, the MathNet team had already prepared the 28k Images Dataset, which contained over 28,000 classified images. Our purpose was twofold: assist in the expansion and release of the 28k Images Dataset, and promote the 28k Images Dataset by releasing papers that used images in the 28k Images Dataset. Our work on the MathNet team can be split into five parts:

1. Implement an image cropper and personally identifiable information (PII) blocker into the ASSISTments workbench to speed up the expansion of the 28k Images Dataset
2. Explore different image clustering methods in order to find common wrong answers in the 28k Images dataset, and release our findings in a paper to motivate other researchers to “beat” us (Pinto, n.d.).
3. Create a crop-resistant invisible watermarking algorithm to apply to images in the 28k Images dataset, to safely share the dataset with researchers without it being leaked
4. Automate the sharing of the 28k Images Dataset
5. Assess the ability of ChatGPT-4 to gather information from images from the 28k Images Dataset

2. Background

This section covers the background information needed to understand the rest of this report. It contains three short general-purpose sections at the beginning, then one detailed section for each of the five projects we worked on.

2.1 ASSISTments

ASSISTments is a free tool developed by the non-profit ASSISTments foundation. Its purpose is letting teachers assign math homework problems online, and letting students submit their answers online (ASSISTments, n.d.-a). In addition to these services, the ASSISTments foundation also harbors 5-10 WPI graduate students, who perform experimental research related to machine learning using the ASSISTments database. One example of such research is QuickComments, which uses machine learning to suggest feedback and scores to teachers grading assignments (ASSISTments, n.d.-b).

2.1.1 MathNet

The MathNet team is a team at ASSISTments headed by Neil Heffernan, Sami Baral, and Li Cheng. According to E-Trials (2023), its purpose is twofold:

1. Release a dataset of 100,000 PII-free images along with papers that use this dataset to encourage the field to increase research in this area.
2. Research and develop different auto-grading methods to grade the images.

2.2 Personally Identifiable Information

Personally Identifiable Information (PII) is information that can be used to identify an individual. Examples of PII include names, faces, and addresses.

2.3 28k Images Dataset

The 28k Images Dataset contains approximately 28,000 images of math homework submissions between first and eighth grade. It has been released by ASSISTments to select researchers.

2.4 Image Cropper and PII Blocker

The Image Cropper and PII Blocker is a tool we developed to allow students to crop images they submitted via the ASSISTments website. This section explains the related terminology we will use.

2.4.1 HTML

HTML, an abbreviation for Hypertext Markup Language, is the language used to describe the structure and general content of a webpage. The language primarily consists of different types of elements denoted by the types of tags used. HTML has become very versatile over the years and is now capable of not only presenting plain text but also images, videos, audio, other multimedia formats, and considerable accessibility support for those who potentially need it. HTML usually works in tandem with other technologies such as Cascading Style Sheets (CSS) for styling and JavaScript for the dynamic interactiveness of the website in order to create a rich and memorable user experience for users.

2.4.2 CSS

CSS, an abbreviation for Cascading Style Sheets, is a declarative styling language that assists in styling the basic HTML elements and components on a web page in order to make a website more aesthetically pleasing and user-friendly. CSS assists in selecting different HTML elements, changing their color, text color, text font, sizing, spacing, and much more. It is also very essential in creating responsive websites that provide seamless and adaptable user experiences on devices of various sizes.

2.4.3 JavaScript

JavaScript is a programming language used in web browsers. It is a versatile programming language that allows developers to make websites dynamic and interactive. JavaScript enables browsers to manipulate the webpage contents of a website, handle user inputs, and send and process asynchronous requests to a server in order to fetch or send data.

2.4.4 Node.js

Node.js is an open-source, cross-platform, server-side JavaScript runtime environment that operates outside of the web browser. If a developer would like to develop both the client-side and the server-side using JavaScript, then they would most likely have to use node for the server-side implementation. Node.js also provides access to a vast and extensive collection of packages and modules to accomplish various functionalities in one's application that have already been developed by others via the Node Package Manager (npm), which helps to significantly expedite the development of new features / functionalities.

2.4.5 Python

Python is a high-level, interpreted programming language that has a wide variety of applications including but not limited to web development, data analysis, machine learning, general scripting, and automation. Using package management systems such as pip or conda, the base functionality of the language can be extended via various libraries and packages. Some packages that were most prevalent and utilized in our work were Pandas, NumPy, SciPy, scikit-learn, PyTorch, TensorFlow, and more.

2.4.6 Axios

Axios is a JavaScript library that is primarily used for making HTTP requests from the client-side (either the web browser or some other Node.js application making the request). It is a popular JavaScript library for front-end web development that assists and simplifies some parts of the process of making asynchronous fetch requests to a server. Axios is able to handle all of the standard fetch request methods such as GET, POST, PUT, and DELETE. Compared to the standard Fetch API built into JavaScript, Axios automatically parses the JSON from the body of the request.

2.4.7 Flask

Flask is a web framework that can be used to build web applications and web APIs using Python. The framework includes features for URL routing and handling of HTTP requests and responses.

2.4.8 YOLOv7

YOLOv7 is one of the latest versions of the You Only Look Once (YOLO) models, a series of computer vision models that far surpasses other state of the art models in terms of speed and accuracy in object detection. The YOLO series of models is a single stage detector which means that it can perform both the classification and the localization in a single pass through the model. YOLOv7 was the model that we used in order to detect the paper containing the student work within the image that they uploaded. This model sent the predicted auto-cropping suggestions for an image to the client-side whenever a request would be made.

2.4.9 Detectron2

Detetectron2 is a state of the art object detection library developed by Facebook AI Research. It performs various computer vision tasks using various object detection and segmentation algorithms in a relatively accurate and efficient manner compared to other object detection libraries.

2.4.10 Canvas API

The Canvas API in JavaScript allows developers to draw lines, shapes, images, objects, and other complex graphics and renders all of it in the web browser. The Canvas API is primarily utilized in conjunction with the <canvas> HTML element which serves as the area where drawings can be made.

2.4.11 MediaDevices API

The MediaDevices API is a Web API that gives developers access to a device's media input devices such as the camera and microphone directly from the browser. This was necessary in order to allow students the ability to take pictures of their work using their device camera within Assistments.

2.4.12 Vue

Vue is a JavaScript framework to simplify the creation of web applications. There are two ways of building a Vue application: the Composition API, and the Options API.

2.4.12.1 Options API

The Options API is one of the two ways to build a Vue application. It is the traditional way to build Vue applications and is easy to understand, but is difficult to use in large applications (Allotey, 2023).

2.4.12.2 Composition API

The Composition API is one of two ways to build a Vue application. It was introduced in Vue 3.0, and aims to address some of the issues in the Options API (Allotey, 2023). It is used by the StudentWorkbench codebase.

2.4.13 StudentWorkbench

StudentWorkbench is the name of the frontend website that students and teachers use to access and submit math problems. It is built using Vue and TypeScript.

2.5 Image Watermarking

Image watermarking is the process of applying a watermark to an image. This section explains the terminology we will use when talking about watermarking.

2.5.1 Invisible Watermarks

An invisible watermark is an image watermark that is not visible to the human eye. It is accomplished by tweaking the color of each pixel in the image in a way that is near-indistinguishable to the human eye.

2.5.1.1 invisible-watermark

invisible-watermark is a Python library for applying an invisible watermark to images. It is developed by GitHub user ShieldMnt and is used by Stable Diffusion, a popular image generation AI.

2.5.1.2 blind_watermark

blind_watermark is a Python library for applying an invisible watermark to images. It is developed by GitHub user guofei9987.

2.5.2 OpenCV

Open Computer Vision Library (OpenCV) is a C++ library with Python bindings that provides many operations for image processing.

2.6 K-Means Clustering

K-means clustering is a fast, popular clustering method with pseudocode as follows:

```
1. Pick k centroids randomly from the dataset.
2. For each image in the dataset:
    a. Find the nearest centroid.
    b. Assign the image to the centroid’s cluster.
3. For each cluster:
    a. Recalculate the centroid. It is the image in the cluster that is closest to all other images.
4. Repeat steps 2-3 until the clusters do not change.
```

In this subsection, we describe the different parts of k-means clustering.

2.6.1 Noise

Noise is any object in an image that is not part of the main content, such as a hand or the lines on a piece of lined paper.

2.6.2 Distance Metrics

K-means clustering requires some way to determine how far apart two items are. The equation to calculate this distance is called the distance metric.

2.6.2.1 Mean Square Error

Mean Square Error uses the following equation to calculate the distance between two images:

$$\frac{1}{N}\sum_{i=1}^{N}\left(X_i - Y_i\right)^2 \qquad (2.6.2.1.1)$$

where:

- N is the number of pixels in the image
- X_i is the value of the pixel at index i in the first image
- Y_i is the value of the pixel at index i in the second image

2.6.2.2 Oriented FAST and Rotated BRIEF

Oriented FAST and Rotated BRIEF (ORB) is an algorithm that finds key points in an image – for example, corners or edges.

2.6.2.3 Template Match

Template matching is the process of finding the regions in a large image that most closely match a small “template” image. We used the OpenCV implementation.

2.6.2.4 CLIP Embeddings

Contrastive Language-Image Pre-Training (CLIP) is an image classification model released by OpenAI. It is able to generate a 512-element vector (embedding) for an image that represents the different features of the image.

2.6.2.5 Angle Between Vectors

The angle between two vectors A and B is defined as

$$\theta = \cos^{-1}\left(\frac{A \bullet B}{|A||B|}\right) \tag{2.6.2.5.1}$$

2.6.3 Edge Images

Edge images are modified versions of images that only show the “edges” in the original image – points where the brightness rises or drops significantly.

2.6.3.1 Canny Edge Detection

Canny Edge Detection is one of the most popular methods of generating edge images.

2.6.3.2 Noise

Noise is small variations in the color of neighboring pixels – for example, slight variations in the color of a piece of paper.

2.7 Dataset Sharing

2.7.1 Google Apps Script

Google Apps Script is a JavaScript-based language developed by Google to enable easy automation of tasks in the Google suite (Drive, Docs, Sheets, etc.) It provides many interfaces for the apps in the Google suite. Examples include allowing users to read / write to a Google document or sheet, send emails, or upload and download files from Google Drive.

2.8 GPT Prompting (Using GPT-4)

GPT-4 is a large language model that can generate both textual and image-based responses to questions that one asks it. For responses that require an image-based response, GPT-4 utilizes the DALL-E 3 model in order to generate an image that corresponds the best with the description that the user has provided in the prompt. GPT-4 also seems to have a new feature that allows it to process images that are uploaded as part of the prompt making it a multimodal language model. It seems to have the ability to extract text from an image fairly accurately and perform further analysis on them. Furthermore, using the Advanced Data Analysis tool, it has the ability to write and run python code natively in the chat via an internal python interpreter and display the resulting output of any graphs, charts, or other visuals produced using popular data science libraries such as pandas, numpy, matplotlib, and seaborn. Additionally, OpenAI has provided an extra mode, ChatGPT Plugins mode, which utilizes the GPT-4 textual model (not multimodal) in combination with various third-party plugins that users can install from the ChatGPT plugins store. The third-party plugins serve to extend the functionality of GPT by allowing it to utilize external services to fetch up-to-date domain-specific information on a certain subject or perform computations, as is the case with the Wolfram Alpha plugin. Wolfram Alpha has a much longer and reliable history of accuracy in performing the correct computations on math problems, particularly ones involving algebraic equations. The plugin also seems to have the capability to generate relevant graphs of equations and number lines accurately which may potentially be valuable in providing supplementary visual aids to students within the primarily textual feedback. GPT-4 also seems to have access to an internal browsing plugin integrated natively in the multimodal version. While this feature did not present an immediate utility for our use case in analyzing middle school students' math work, it can potentially be useful if GPT-4 does not have knowledge about some obscure math topic, allowing it to consult external sources in order to provide a more informed response.

3. Methodology

3.1 Overview

Our work was separated into five parts: the image cropper and PII blocker, image clustering, image watermarking, dataset sharing, and GPT prompting. This section details our methods for each of these parts.

3.2 Image Cropper and PII Blocker

The image cropper and PII blocker were two features that attempted to solve the same problem: minimizing the amount of PII that was showing in each image. We implemented them via the Insert Image and Take Photo buttons on the ASSISTments website. This section covers their implementations.

3.2.1 Purpose

The purpose of the image cropper and PII blocker was to minimize the amount of PII shown in each image. This makes the images easier to release in a dataset, since each image in the dataset has to be manually verified to not contain PII: rather than have the verifiers crop the images and run Detectron themselves, the image cropper and PII blocker could do that automatically.

3.2.2 Methods

The image cropping feature can be utilized by the students through two different ways of submitting their work. Students can submit an image of their work in one of two primary ways: they can upload their work via the "Insert Image" option which allows them to upload an image that is stored on their device and then crop it as they would like to using the cropping tool and a "Take Photo" option that allows them to take a photo using their device camera which will automatically crop based on a predefined bounding box. We developed the image cropper feature in two phases: a proof of concept demonstrated on the demo app and API that were passed on to us from previous members working on the project and later on in the official StudentWorkBench interface for Assistments. The demo application was a Vue.js / TypeScript frontend interface that would simulate the StudentWorkBench frontend for testing purposes. The PII Blocker was a Flask API server that ran in the backend and processed images that would arrive from the client-side by applying black segmentation masks over the regions of the image that would be predicted to contain PII. The production StudentWorkbench codebase incorporates a frontend web application developed primarily with Vue.js and TypeScript similar to the demo application.

3.2.3 Insert Image Option

The insert image option was already relatively fleshed out in the demo application we inherited. Our job was to add additional polish, as well as rework the UI to meet the StudentWorkbench codebase standards. We ran into three major issues:

1. The demo code was written using the Options API. However, the StudentWorkbench codebase uses the Composition API, so we had to convert the demo code to the Composition API.

2. The demo code was written in vanilla JavaScript, while the StudentWorkbench codebase uses TypeScript. In order to keep the StudentWorkbench codebase consistent, we had to convert the demo code to TypeScript.
3. We were unable to use the image cropping API as the costs for hosting a server were too high. As a result, we had to remove all references to the API from our code in StudentWorkbench.

3.2.4 Take Photo Option

The Take Photo option was first implemented in the demo application for prototyping purposes. In order for this submission option to work, the application would need access to the user's device camera. This was primarily done using the MediaDevices API which streamlined the process of acquiring permission and access of the user's live video feed from the camera. The live video feed was then displayed using the <video/> HTML element on the web page. Using CSS, a <div> element was styled to look like a rectangular bounding box. This bounding box represented the dimensions to which the originally taken image would be cropped. The bounding box was superimposed on top of the live camera feed in order to indicate to the students that they should place their work within this area in order to facilitate the capture of as much of their work as possible in the image, while reducing the amount of unnecessary background as possible. When the student is satisfied with the positioning of their work and decides to take the photo, the photo is immediately cropped based on the ratios of the video element dimensions and the bounding box dimensions as well as the relative positioning of both elements to their parent container which was also a <div> element. The resulting cropped image is drawn to a HTML <canvas> element using the Canvas API for the student to preview before deciding to either go forward with it or retake it. If the student decides to retake it, they are returned to the take photo dialog pane. Otherwise, the student is taken to a new pane where a HTTP request is made to the PII Blocker. When the PII blocker returns the PII-blocked image, it will be displayed to the student as a preview. They can choose to return to the Take Photo pane again if they feel it necessary to retake the photo again (perhaps due to their work being covered in the PII-blocked image) or they can continue with the PII-blocked image if they are satisfied with the results. Finally, the student can choose to insert the cropped and PII-blocked image in the open response box if they are satisfied with the quality, return to any previous step, or cancel the whole Take Photo process entirely.

3.2.5 Flaws and Limitations

Some flaws and limitations of the "Take Photo" option is that there might be a reduction in the resolution of the cropped image that contains the student work. Since the bounding box occupies only a limited portion of the total camera feed, the resolution and quality of the final cropped image will suffer slightly. Another limitation of the "Take Photo" option would be that there is generally a high probability that some small residual PII still remains in the cropped image due to students partially capturing the hand that they are using to hold up their work while taking the photo. A potential flaw of the current "Take Photo" implementation would be that if the aspect ratio of the bounding box does not happen to match the aspect ratio of the preview canvas for the cropped image, the cropped image can become severely distorted. Additionally, the bounding box shifts and distorts to an unexpected location if the browser window is resized to be very small.

3.3 Image Clustering

We clustered images within the 28k Images Dataset using k-means clustering, which separates all the images we give it into *k* groups, called clusters. The k-means clustering algorithm requires a distance metric, which is a way to numerically express how closely two images are related. We tested four different distance metrics:

1. Mean Squared Error
2. Oriented FAST and Rotated BRIEF
3. Template Match
4. Angle between CLIP Embeddings

We also tested four different preprocessing effects on the images:

1. No preprocessing: the control group. No preprocessing effects were applied to the images.
2. Cropped images: cropping the images to only the relevant content.
3. Edge images: applying Canny edge detection to transform the images into edge images.
4. Cropped edge images: First cropping the images, then applying Canny edge detection to transform them into edge images.

In this subsection, we describe how we implemented k-means clustering with the different preprocessing effects and distance metrics, along with how we tested and scored our implementations.

3.3.1 Purpose

One of the goals of the MathNet project was to research and develop different auto-grading methods to grade the images (E-Trials, 2023). We were given the task of exploring image clustering to find CWAs. The theory was that similar images will be put into the same cluster, which can help us identify CWAs. When we published our results in a paper, we hoped to motivate other researchers to “beat” us (Pinto, n.d.).

3.3.2 Clustering Algorithm

We chose k-means clustering since it was by far the most common clustering algorithm we found. It has one parameter we can manipulate: the distance metric, which determines how close (or far away) two images are.

3.3.3 Preprocessing Effects

We tested three different preprocessing effects on our images: converting the images to edge images, cropping the images to the smallest size that still contained the relevant content, and combining the crop with the edge images. We also tested a control, “No Preprocessing”, where no preprocessing effects were applied to the images. In this subsection, we detail our implementations of the three different preprocessing effects.

3.3.3.1 Edge Images

Many of our images have background noise or other unnecessary content. By creating edge images, we hoped to remove some of this noise to allow our distance metrics to work better.

Figure 3.3.3.1.1 shows an example of an image that was converted to an edge image. Through trial and error, we found a six-step process to generate clean edge images:

1. Apply k-means clustering (k=11) on the pixels within the image using `cv2.kmeans()`. This reduces the image down to 11 colors, which removes most of the noise.
2. Convert the image to grayscale using `cv2.cvtColor()` with `cv2.COLOR_BGR2GRAY`. This is necessary for Canny edge detection to run.
3. Increase the contrast of the image. This lets Canny pick up thin, light pencil lines.
4. Apply a Domain Transform Filter using `cv2.ximgproc.createDTFilter`. This is a fancy denoising algorithm that removes most noise from the image.
    a. We used $\sigma_{spatial} = 30$, $\sigma_{color} = 10$, DTF_IC, and 5 iterations.
5. Apply a gravitational blur. This is similar to a Gaussian blur, but it preserves edges better.
6. Run the Canny edge detection algorithm on the images using `cv2.Canny`. This is what actually creates the edge images. We used the following parameters:
    a. A lower threshold of $max(0,\ 0.66\ *\ median(image))\ /\ 3$.
    b. An upper threshold of $max(min(255,\ 1.33\ *\ median(image))\ /\ 7,\ 10)$.
    c. An aperture size of 3.
    d. No L2 gradient.

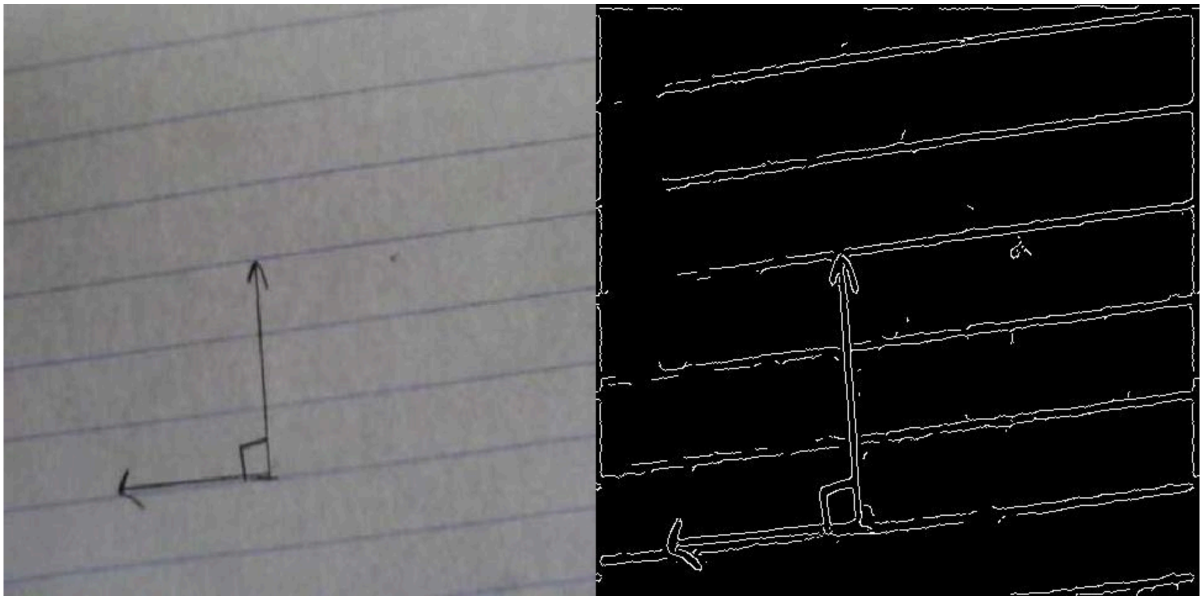


Figure 3.3.3.1.1: An unprocessed image (left) and its corresponding edge image (right).

3.3.3.2 Cropped Images

Many of the images have their relevant content in the center of the image, and large areas on the sides where there is no relevant content. Cropping the images removes the large areas of irrelevant content, which allows our distance metrics to perform better. See Figure 3.3.3.2.1 for an example of a cropped image.

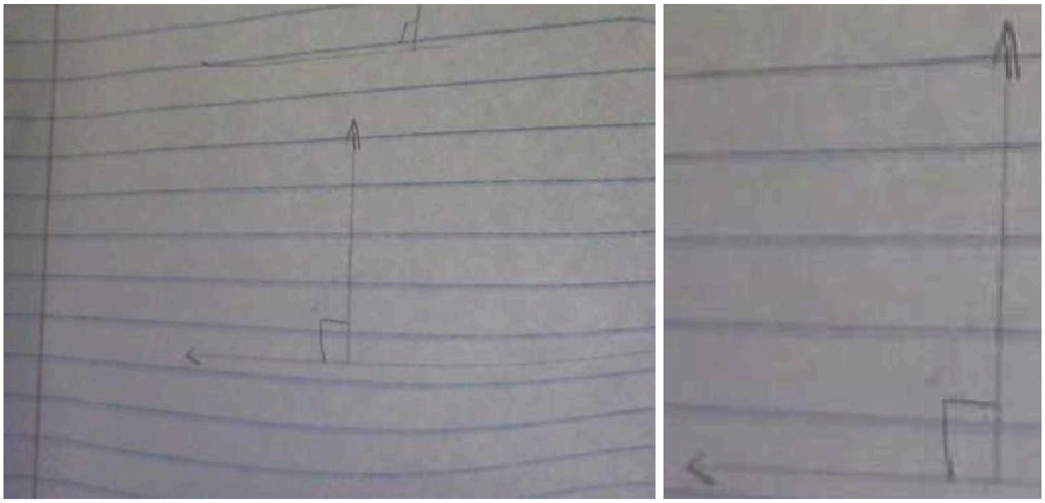


Figure 3.3.3.2.1: An unprocessed image (left) and its corresponding cropped version (right).

3.3.3.3 Cropped Edge Images

We created cropped edge images by first cropping the image using the method detailed in Section 3.3.3.2, then applying the edge image method detailed in Section 3.3.3.1. Figure 3.3.3.3.1 shows an example result from this method.

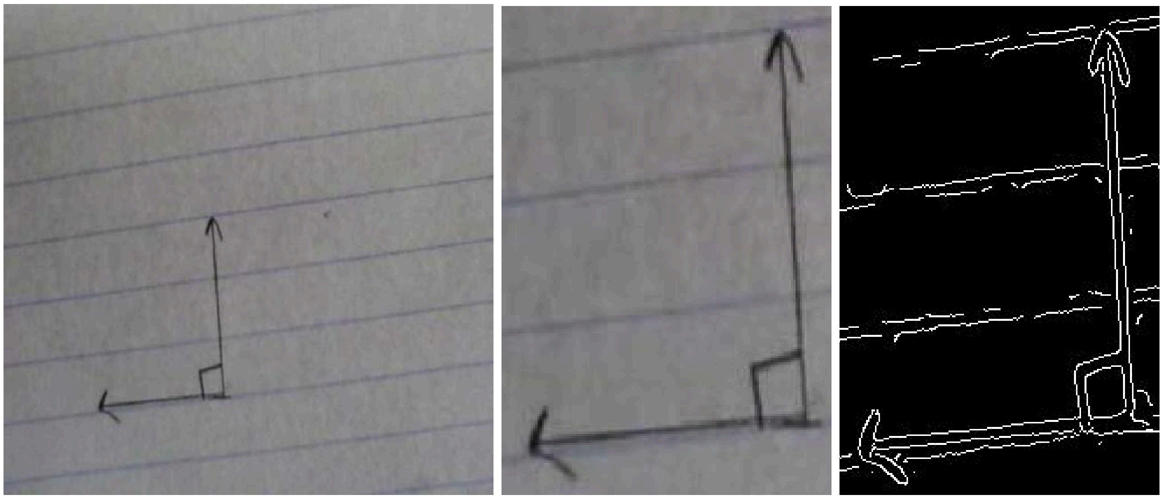


Figure 3.3.3.3.1: An unprocessed image (left), its corresponding cropped version (middle), and the final cropped edge image (right).

3.3.4 Distance Metrics

K-means clustering requires a distance metric: given two images, we need to generate a number that represents the distance between them. For example, if we were to cluster a group of 2D points, we could use the Euclidean distance between them as a distance metric

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.3.4.1}$$

In our research, we found four distance metrics: Mean Squared Error (MSE), Oriented FAST and Rotated BRIEF (ORB), OpenCV’s Template Match, and the angle between two CLIP-generated embeddings.

3.3.4.1 Mean Squared Error

We take the difference between each individual pixel in each of the images and square it to yield an overall difference between the images (See Eq. 2.6.2.1.1). Figure 3.3.4.1.1 shows two very different images with high MSE, and Figure 3.3.4.1.2 shows two very similar images with low MSE.

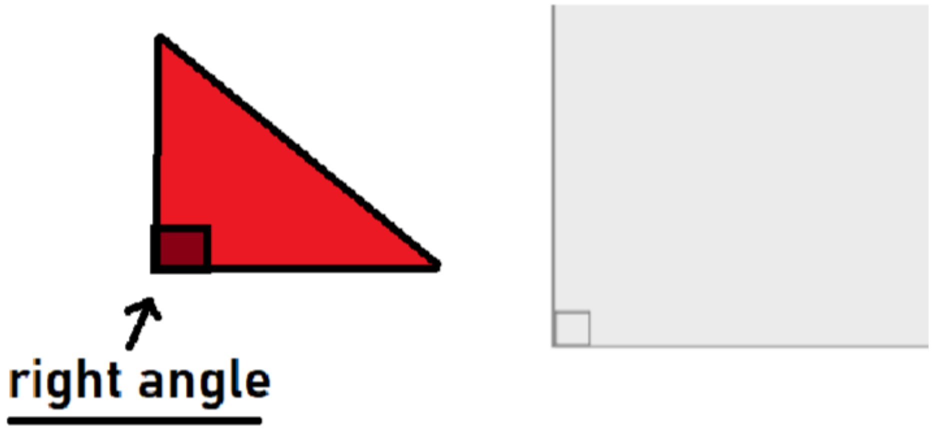


Figure 3.3.4.1.1: Two very different images with a MSE of 2531.

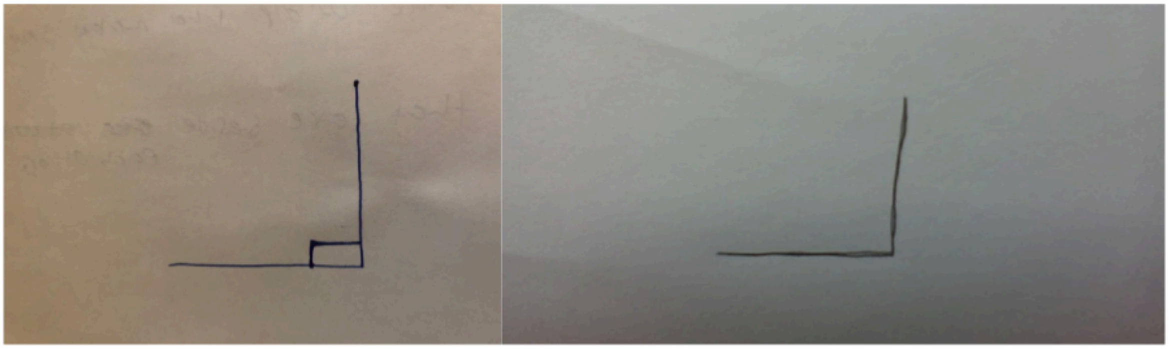


Figure 3.3.4.1.2: Two very similar images with a MSE of 53.25.

3.3.4.2 Oriented FAST and Rotated BRIEF (ORB)

We compared the key points generated by ORB using OpenCV's BFMatcher with normalized Hamming distance (cv2.NORM_HAMMING). We then used the percentage of correctly matched keypoints as our distance metric. Figures 3.3.4.2.1 and 3.3.4.2.2 show two examples of ORB comparison between two images.

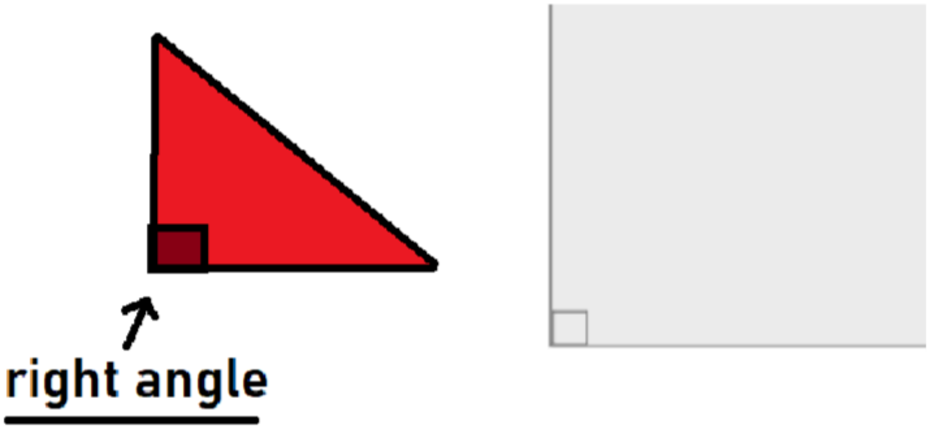


Figure 3.3.4.2.1: ORB comparison between two very different images with a score of 0.2.

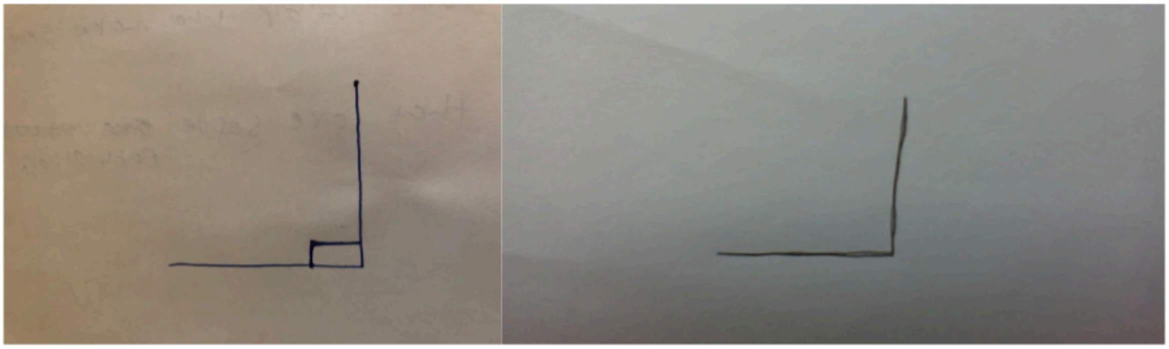


Figure 3.3.4.2.2: ORB comparison between two very similar images with a score of 1.

3.3.4.3 Template Match

For two images A and B, we use Template Match in the following fashion:

$d = min(match(A\ onto\ B),\ match(B\ onto\ A))$.

We match A onto B, then B onto A, and find the resulting match scores. We take the minimum of the two results since there are some cases where A matches closely onto B, but B does not match closely onto A. For example, if A is an image of a square and B is an image of a blank sheet of graph paper, A will match closely onto B but B will not match closely onto A.

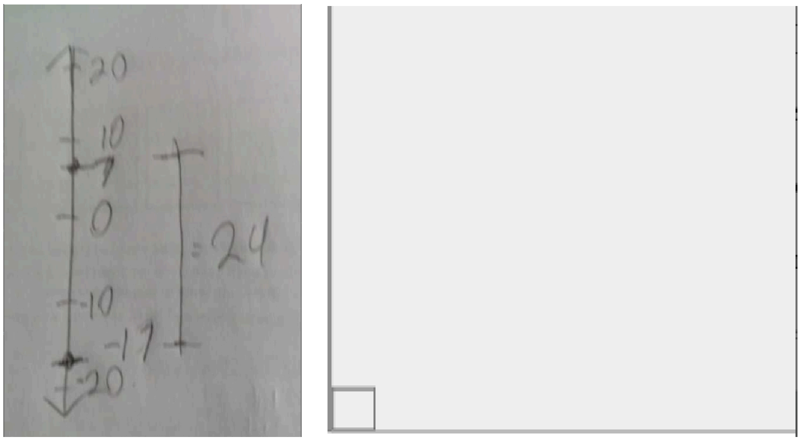


Figure 3.3.4.3.1: Template Match comparison for between two very different images with a score of 0.25.

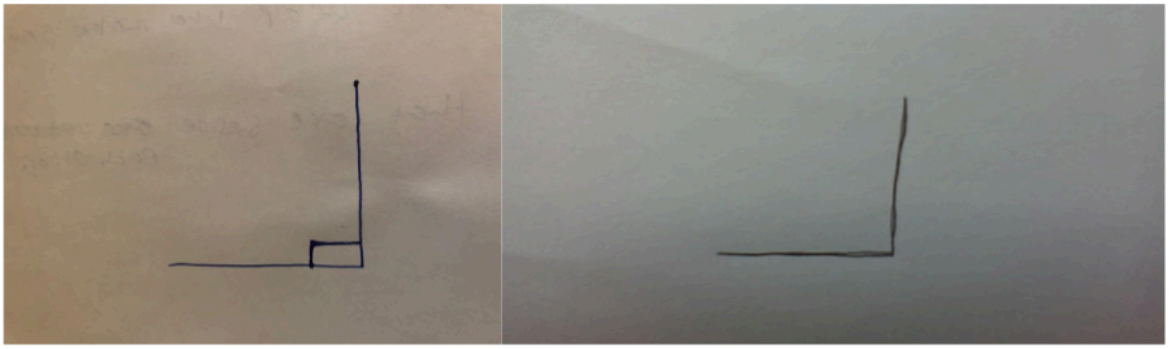


Figure 3.3.4.3.2: Template Match comparison for between two very similar images with a score of 0.98.

3.3.4.4 CLIP Embeddings

As our distance metric, we used the angle between the embeddings generated between each image (See Eq. 2.6.2.5.1). We used the largest model CLIP offers, ViT-L/14@336px. Figures 3.3.4.4.1 and 3.4.4.4.2 show the scores for comparing two very similar and two very different images, respectively.

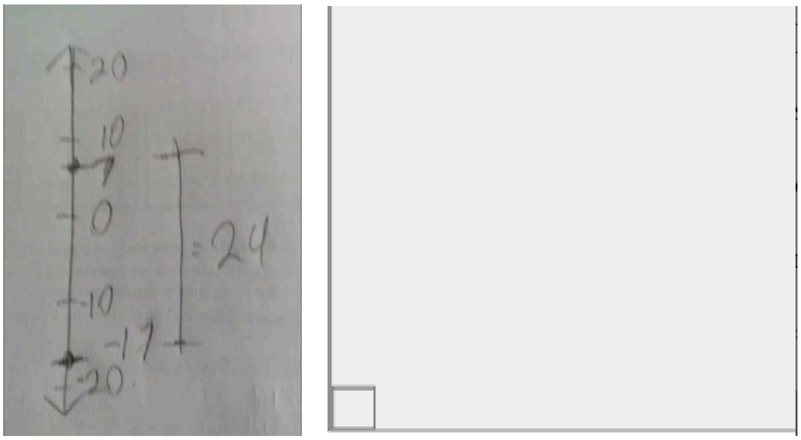


Figure 3.3.4.4.1: CLIP comparison for between two very different images with a score of 2.49.

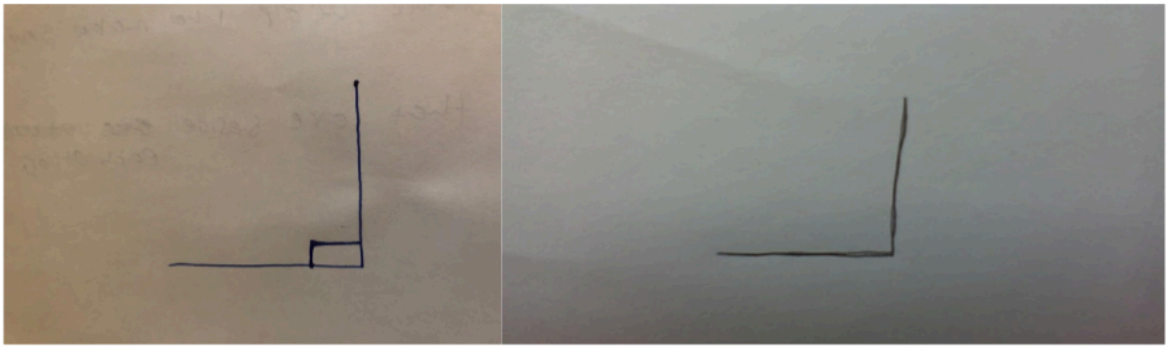


Figure 3.3.4.3.2: CLIP comparison for between two very similar images with a score of 2.53.

3.3.5 Dataset Selection

We selected two visually distinct problems from the 28k Images Dataset. The problem text was as follows:

Problem 1: Draw a right angle.

Problem 2: This diagram (Figure 3.3.5.1) represents 3 batches of light yellow paint. Draw a diagram that represents 1 batch of the same shade of light yellow paint.

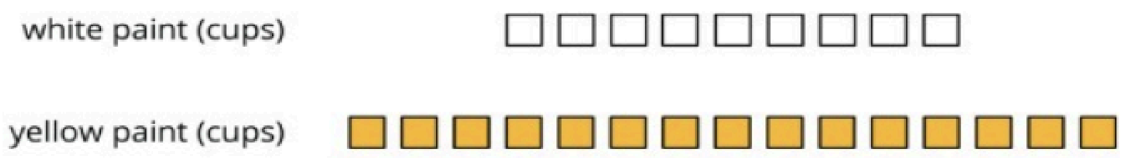


Figure 3.3.5.1: A diagram showing the paint required to make three batches of light yellow paint.

Our dataset contained 318 images, 159 from each problem. Problem 2 had 269 images in total, so we randomly sampled 159 in order to balance out the dataset.

3.3.6 Performance Metric

We measured our distance metrics' accuracy by comparing the number of images from each problem in each cluster. If a cluster has 40 images from problem 1 and 40 images from problem 2, we can conclude that the distance metric was unable to differentiate between the two problems. However, if a cluster has 75 images from problem 1 and 5 images from problem 2, we can conclude that the distance metric was able to differentiate between the two problems.

3.3.7 Flaws and Limitations

Each distance metric we described above has its own flaws, which this subsection will cover.

3.3.7.1 MSE

MSE has two limitations that stem from it comparing raw pixel values:

1. It cannot handle scaled or rotated images.
2. It is heavily influenced by background noise.

3.3.7.2 ORB

ORB has one flaw: currently, there is no good way to compare the keypoints compared for both images. In order to do this, we would have to develop a new algorithm.

3.3.7.3 Template Match

The OpenCV implementation of Template Match, which we used, does not support rotated or scaled images. However, other implementations of Template Match do support rotated and scaled images, so in the future those implementations could be used as a distance metric.

3.3.7.4 CLIP

Due to being a machine learning model, CLIP is incredibly slow to run on a CPU, taking upwards of 10 seconds to process each image while a GPU averages 0.1 seconds per image – two orders of magnitude faster.

3.4 Image Watermarking

We were not confident that there was no PII in the 28k images dataset, so we decided to only release the dataset to select researchers. Since we did not want any of the researchers to leak the dataset, we applied an invisible watermark to images. We found two libraries for invisible watermarking, and ran a series of robustness tests on each one. We also modified one of the libraries to support decoding an image after it was cropped.

In this section, we describe the specifics of our robustness tests along with how we implemented a crop-resistant invisible watermark.

3.4.1 Purpose

Although Detectron2 had been run on all of the images in the 28k images dataset, we were not confident that all PII had been removed from the images. Detectron2 only blocks out people, so if there is PII on the paper (such as a name) it will not be blocked out. Because of this, we decided to only release the dataset to select researchers. There is still the possibility of a researcher leaking the dataset, so we decided to apply an invisible watermark to images in the dataset to further discourage leaking. The watermark had to be invisible since visible watermarks on images can easily be removed.

3.4.2 Methods

In our research, we found two existing invisible watermark libraries: invisible-watermark and blind_watermark. Since the images can be modified (for example, cropped) by a researcher before they are leaked, it is important to have a robust watermark so we can correctly identify who leaked an image.

3.4.3 Robustness Tests

Both libraries advertised similar robustness capabilities, so we devised a series of tests to determine which one to use. These tests were pulled from the invisible-watermark GitHub page and cover compression, noise, and masking, among other things. We added one test, a severe crop, where the image is cropped on all sides. As Figure 3.4.3.1 shows, some images do not have the student's solution in the main focus, so it is likely they will be cropped on all sides (see

Figure 3.4.3.2 for an example crop). Having a watermark that is resilient to cropping allows us to correctly identify which researcher leaked an image, even if the image has been modified.

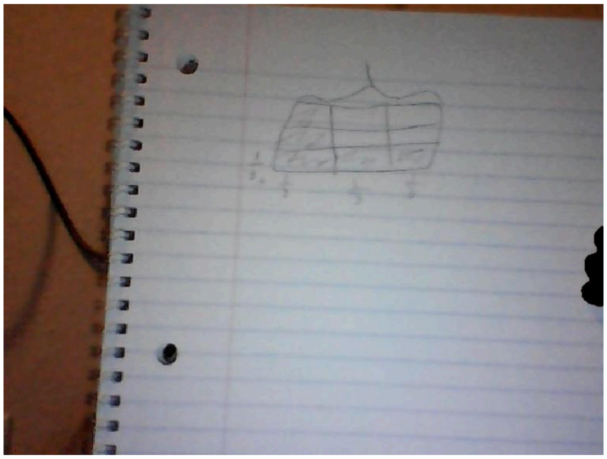


Figure 3.4.3.1: An image that will probably be cropped on all sides.

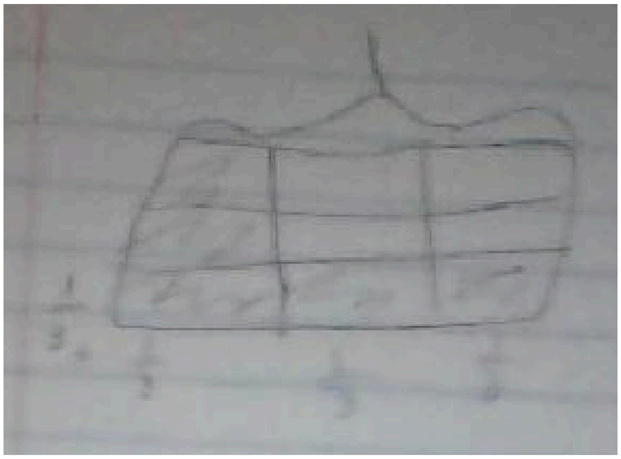


Figure 3.4.3.2: The same image as Figure 1, cropped on all sides.

The final suite of robustness tests was as follows:

- JPG Compress
- Add Noise
- Decrease brightness
- Overlay (draw things on the image)
- Mask (block out sections)
- Mild crop: remove ~50px from the right side
- Severe crop: remove ~50px from all sides
- Resize to 50% of the original size
- Rotate clockwise by 30°

Both invisible-watermark and blind_watermark provided similar results, although invisible-watermark incorrectly decoded one character on both image compression and mask (see Table 4.4.1.1). Both libraries failed to decode the images under both mild and severe crops, which was not ideal. In order to circumvent this issue, we decided to modify the invisible-watermark code to be more resilient to cropping. We chose invisible-watermark since its code is documented in English, while blind_watermark is documented in Chinese, which neither of us can read.

3.4.4 The invisible-watermark Algorithm

In order to make the invisible-watermark code more resilient to cropping, we first needed to understand how the invisible-watermark code encodes and decodes an image. The image is first converted to the YUV format, and ASCII representation of the message is converted into a string of bits. Then, each bit is encoded into an 8px by 8px square in the image. Figure 3.4.4.1 shows the order in which the bits are encoded.

When invisible-watermark decodes an image, it reads the bits in the same order it encoded it. Since the message is encoded multiple times in the image, the values for each bit in the message are averaged – for example, if the first bit was read as 0, 1, 0, and 0, then it would be averaged to 0.

The encoding scheme in invisible-watermark is vulnerable to all forms of crop, even a small 1-pixel one. As Figure 3.4.4.2 shows, the bits in the image will be read in the wrong order: 0-1-2-3, 0-1-2-3, 0-1-2-0, 1-2-3-0, and so on. This will mess up the bit averages, and results in a garbled message.

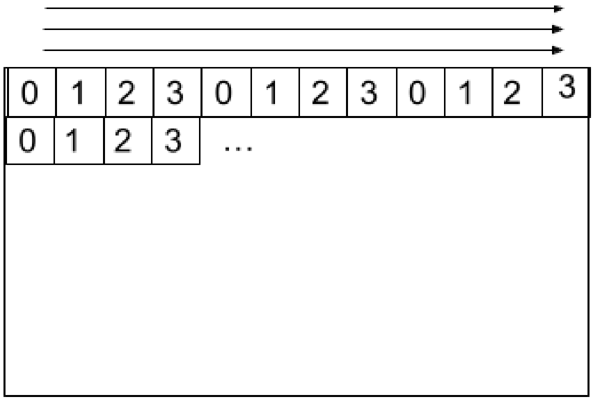


Figure 3.4.4.1: The bit encoding scheme used by invisible-watermark, with an example 4-bit message. The bits are encoded in a left-to-right order. When the image is decoded, the bits are read 0-1-2-3, 0-1-2-3, 0-1-2-3, and so on.

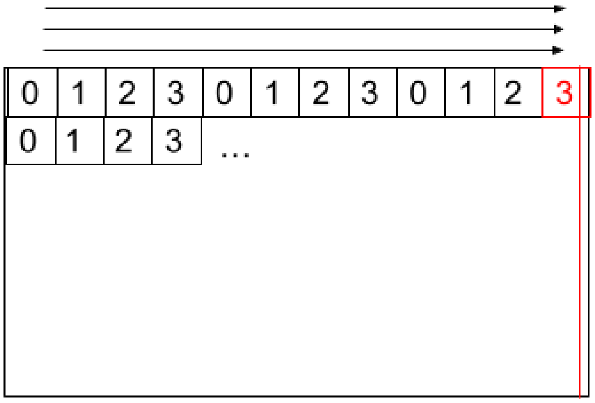


Figure 3.4.4.2: The bit encoding scheme from Figure 3.4.4.1, after a mild crop removes one pixel from the right side of the image. The decoded bits are read 0-1-2-3, 0-1-2-3, 0-1-2-0, 1-2-3-0, and so on.

3.4.5 Our invisible-watermark Changes

In order to solve the cropping problem, we decided to encode the bits in a different order. We accomplished this by encoding the bits into blocks (see Figure 3.4.5.1), which makes the encoding much more resilient to cropping (see Figure 3.4.5.2).

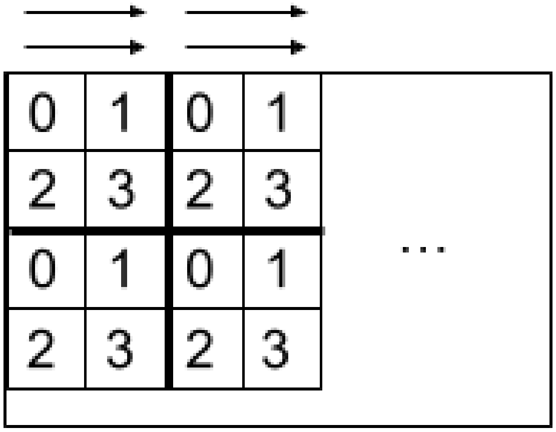


Figure 3.4.5.1: The bit encoding scheme used by our modified version of invisible-watermark, with an example 4-bit message. The bits are encoded in a square pattern, so they are much

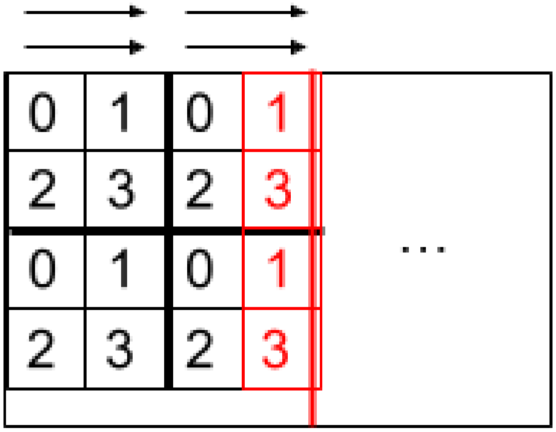


Figure 3.4.5.2: The bit encoding scheme from Figure 3.4.5.1, after a severe crop removes a large amount of pixels from the right side of the image. The decoded bits are read 0-1-2-3,

more resilient to cropping. The decoded bits are read 0-1-2-3, 0-1-2-3, and so on.

0-?-2-?, 0-1-2-3, 0-?-2-3, and so on.

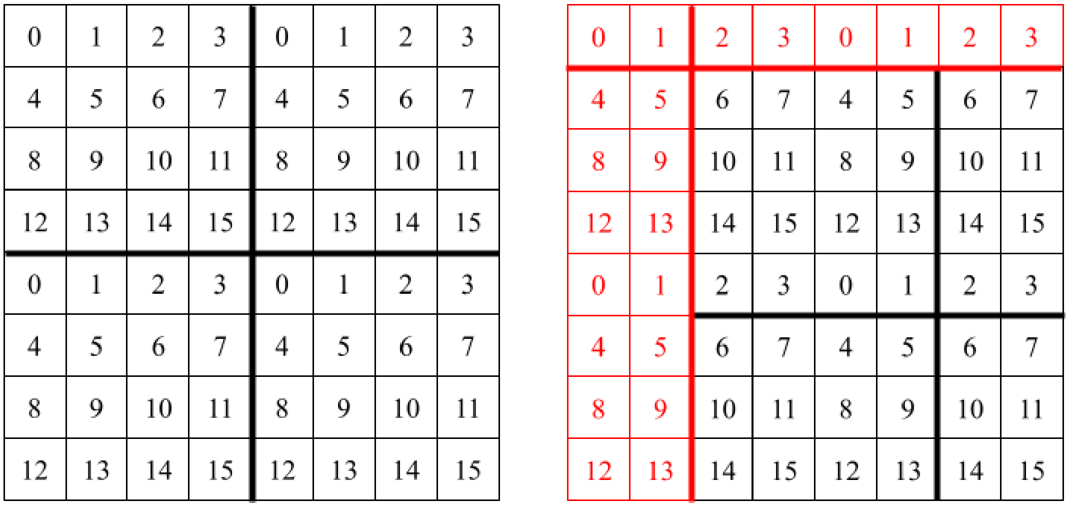
3.4.6 Decoding a Cropped Image

This subsection details the three issues we run into when decoding a cropped image.

3.4.6.1 Grid Starting Point

We can think of each encoded image as a grid of encoded bits, where each index in the grid is 8x8 pixels.

When an image is cropped, it will not always be cropped by a multiple of 128 pixels, so the grid may be read out of order. To solve this, we test every possible ordering of the bits. For example, the bit string “10011” would be tested as “10011”, “11001”, “11100”, “01110”, and “00111”.



0	1	2	3	0	1	2	3
4	5	6	7	4	5	6	7
8	9	10	11	8	9	10	11
12	13	14	15	12	13	14	15
0	1	2	3	0	1	2	3
4	5	6	7	4	5	6	7
8	9	10	11	8	9	10	11
12	13	14	15	12	13	14	15

0	1	2	3	0	1	2	3
4	5	6	7	4	5	6	7
8	9	10	11	8	9	10	11
12	13	14	15	12	13	14	15
0	1	2	3	0	1	2	3
4	5	6	7	4	5	6	7
8	9	10	11	8	9	10	11
12	13	14	15	12	13	14	15

Figure 3.4.6.1.1: A comparison between the grids of an uncropped (left) and cropped (right) image. In the cropped image, the cropped off portions are denoted in red.

See Figure 3.4.6.1.1 for a comparison between the grid of a cropped image versus an uncropped image. In the uncropped image, the grid is read as:

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix}$$

This is the correct ordering of pixels.

In the cropped image, the grid is read as:

$$\begin{bmatrix} 6 & 7 & 4 & 5 \\ 10 & 11 & 8 & 9 \\ 14 & 15 & 12 & 13 \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

This is incorrect. In order to fix this, we need to shift all elements right by two rows:

$$\begin{bmatrix} 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

And shift all elements down by one row:

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix}$$

This is the correct ordering of elements in the grid, and the message will be decoded correctly. In the actual decoder, we need to test all 64 possible combinations of right and down shifts on our 8x8 matrix.

3.4.6.2 Pixel Starting Point

A non-cropped image is very simple to decode: decode the 8x8 square starting at (0, 0), move right to decode the square starting at (8, 0), and so on.

When we decode a cropped image, we do not know how many pixels were cropped off, so we cannot determine the correct starting point. In order to solve this, we treat each pixel in the square from (0, 0) to (7, 7) as the starting pixel.

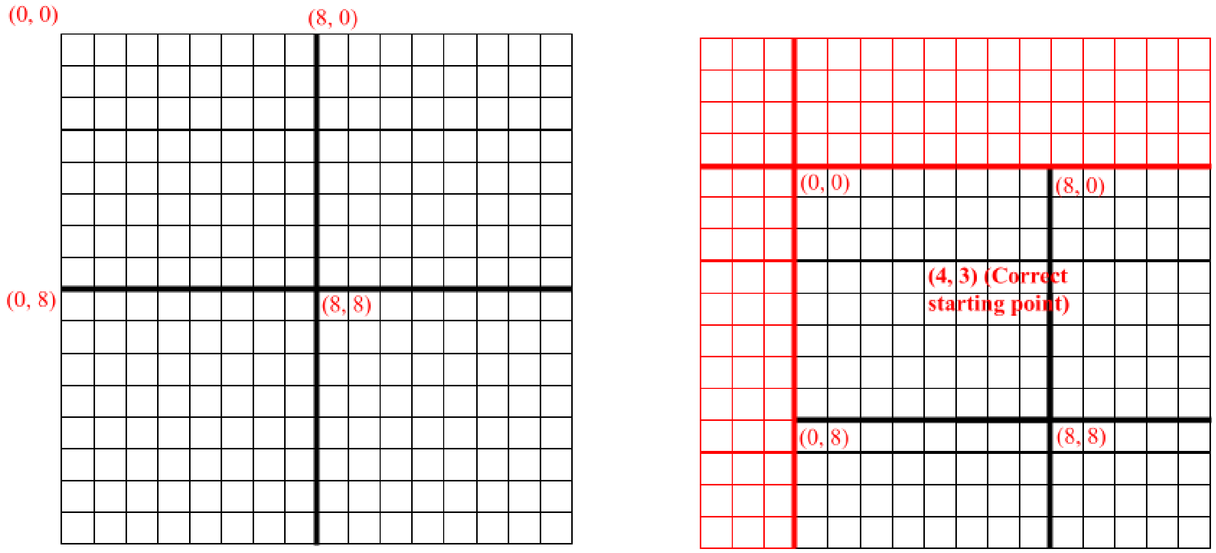


Figure 3.4.6.2.1: A comparison between the process of decoding an uncropped (left) and a cropped (right) image. In the cropped image, the cropped off portions are denoted in red.

For a comparison between the cropped and uncropped process, see Figure 3.4.6.2.1. In the uncropped image, we start at (0, 0) and decode the bit from the 8x8 square there. In the cropped image, four pixels have been removed from the top and three from the left, so we need to treat the pixel at (4, 3) as the starting point for the grid. In the actual decoder, we have no way of knowing that (4, 3) is the correct starting point, so we have to test all 64 pixels from (0, 0) to (7, 7).

3.4.6.3 Key Character

The grid and pixel starting point tests will result in many garbage outputs, since there are inevitably going to be some pixel combinations that result in a text solution that is not the message that was encoded. For a single image there can be upwards of 100 garbage outputs, making it difficult and time-consuming to find the correct one. In order to remove many of these outputs, we added a “key” character to the beginning of each string: ‘_’. If a string we decode does not have this key character at the start, we know that it is not the correct string and can discard it.

3.4.7 Flaws and Limitations

The watermarker has multiple flaws:

1. It cannot encode images smaller than 128x128 pixels. A 32-character message uses 256 bits, which forms a 16x16 grid. Multiply that by the 8x8 pixel grid required to encode each bit, and we get the 128x128 image.
2. The decoding of an image may fail even if the encoding succeeds.
3. There are several transformations the decoder fails to decode, for example resizing and rotation.
4. The message must be exactly 32 characters long.

3.5 Dataset Sharing

We automated the data sharing process for the 28k Images Dataset since we were giving a uniquely watermarked copy of the dataset to each researcher. Watermarking is a tedious, time-consuming process that can be easily automated. In this section, we describe our automation process and implementation.

3.5.1 Purpose

Sharing the dataset with another researcher is a tedious process. Because each researcher needs a unique copy of the dataset with a unique watermark, the entire dataset needs to be copied over to a new folder, which takes hours. After that, we also have to download, watermark, and reupload one random image, then manually log which image we watermarked and the message we watermarked it with. Overall this is a very time-intensive process that is much better automated.

3.5.2 Process

There are a wide variety of form submission tools, but we decided to go with Google Forms since it can be integrated with Google Sheets: all form submissions can be put into a spreadsheet automatically. We devised a six-step process for watermarking the images. Each step is preceded by “Frontend” or “Backend”, which signifies which one is executing the step.

1. **Frontend**: When a new request is submitted, upload the request’s data to a Google Sheet and add in a cell that marks whether or not the request has been verified. Then, notify an ASSISTments verifier. The verifier will look over the request to make sure that the requester is legitimate. If the verifier approves the request, they will change the “Verified” cell from “0” to “1”.
2. **Frontend**: When a “Verified” cell is changed to 1, send a GET request to the backend containing necessary watermarking information such as the researcher’s name, which we use as the message when watermarking the images.
3. **Backend**: Once a GET request is received from the frontend, make a copy of the dataset folder. Within the copied folder, watermark one random image.

4. **Backend**: Compress the copied folder into a ZIP file and upload it to Google Drive. Append a new row to a prespecified Google Sheet that contains the name of the image that was watermarked. Upload the image that was watermarked to Google Drive to keep a backup of it.
5. **Frontend**: When a new row is added to the Google Sheet the backend uploads to, share the ZIP file with the researcher.
6. **Frontend**: After one week, delete the ZIP file from Google Drive to save storage space.

3.6 GPT Prompting

When the image analysis feature was released on GPT-4, it showed promising signs that it could potentially assist with solving and explaining more open-ended math problems. We set out to test its capabilities.

3.6.1 Purpose

Utilizing the collective features of GPT-4, we wanted to test its capabilities up until now on math problems that would typically be presented to middle schoolers as well as assessing its responses to some student answers themselves. Another related inquiry that we wanted to pursue was to assess how much more information beyond basic optical character recognition (OCR) GPT-4 was capable of gathering from the submitted math work images. More specifically, we wanted to assess what GPT would gather from diagrams, hand-drawn images, graphs, number lines, etc. We wanted to assess the interpretability and accuracy of the explanations that GPT produced.

3.6.2 Initial Prompting Approach and Testing

To begin with, GPT was presented with a math problem description followed up in subsequent prompts with images of student answers so that it could provide feedback on them. The first problem that was tested was problem ID 1754531. The problem description was as follows: "Sketch the unique triangles that can be made with angle measures 40° and 100° and side length 3. How do you know you have sketched all possibilities?" This problem was chosen in order to test GPT's capabilities in understanding images involving diagrams. For this problem, the diagram happened to be of an isosceles triangle. The prompts that were provided to GPT for this problem are as follows sequentially:

1. You are a teacher who is providing feedback to students on responses to their math work. The image contains some material that is not relevant to the question that you will be assessing. The question that you will be assessing is "Problem Text: Sketch the unique triangles that can be made with angle measures 40° and 100° and side length 3. How do you know you have sketched all possibilities?" The student answer is near and around the question. First, describe/summarize the student response. Then, assess the correctness of their response.

   (See Appendix B1 for the image attached to this prompt)

The image attached to prompt #1 and the second prompt onwards can be found in Appendix B1 for this problem.

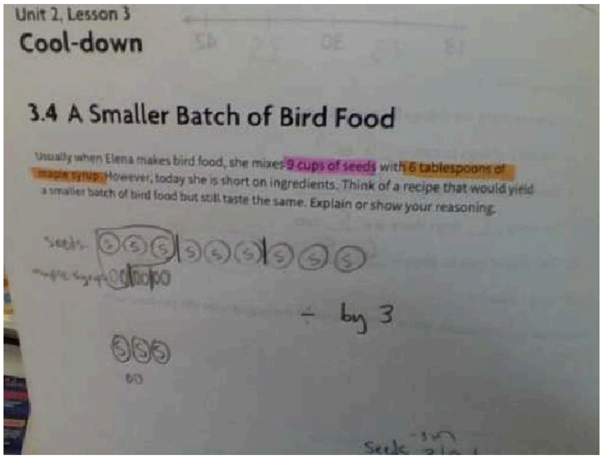
The second problem that was tested was problem ID 1755170. The problem description is as follows: "Graph your equation using graphing technology. Mark the point on the graph that represents the amount after 3 weeks." This problem was chosen in order to test GPT's ability in understanding images of graphs. This was selected as it is a fairly challenging type of problem to assess with lots of small details that have to be verified for an answer to be considered correct such as the student labeling the axes properly, incrementing the axes properly, identifying the slope and equation of the line or sequences of points marked that would denote a line, and providing a clear indication of the point after 3 weeks. The prompts that were provided to GPT for this problem are as follows sequentially:

1. You are a teacher who is providing feedback to students on responses to their math work. The image contains some material that is not relevant to the question that you will be assessing. The question that you will be assessing is "Graph your equation using graphing technology. Mark the point on the graph that represents the amount after 3 weeks." The student answer is near and around the question. First, describe/summarize the student response. Then, assess the correctness of their response.
   (See Appendix B2 for the image attached to this prompt)

The image attached to prompt #1 and the second prompt onwards can be found in Appendix B2 for this problem.

The third problem that was tested was problem ID 1622991. The problem description is as follows: "Usually when Elena makes bird food, she mixes 9 cups of seeds with 6 tablespoons of maple syrup. However, today she is short on ingredients. Think of a recipe that would yield a smaller batch of bird food but still taste the same. Explain or show your reasoning." This problem has no initial diagram or graph that the student can modify or build off. It is up to the student how they would like to explain their answer or what diagrams they would like to specifically make. This problem tests GPT's consistency in providing feedback for varying supporting diagrams between different students. The prompts that were provided to GPT for this problem are as follows sequentially:

1. You are a teacher who is providing feedback to students on responses to their math work. The image contains some material that is not relevant to the question that you will be assessing. The question that you will be assessing is "Usually when Elena makes bird food, she mixes 9 cups of seeds with 6 tablespoons of maple syrup. However, today she is short on ingredients. Think of a recipe that would yield a smaller batch of bird food but still taste the same. Explain or show your reasoning." The student answer is near and around the question. First, describe / summarize the student response. Then, assess the correctness of their response.



Unit 2, Lesson 3

Cool-down

3.4 A Smaller Batch of Bird Food

Usually when Elena makes bird food, she mixes 9 cups of seeds with 6 tablespoons of maple syrup. However, today she is short on ingredients. Think of a recipe that would yield a smaller batch of bird food but still taste the same. Explain or show your reasoning.

Seeds
maple syrup
÷ by 3
Seeds

Figure 3.6.2.1: A student answer of problem ID 1622991 that contains the problem description.

2. Here is another student response

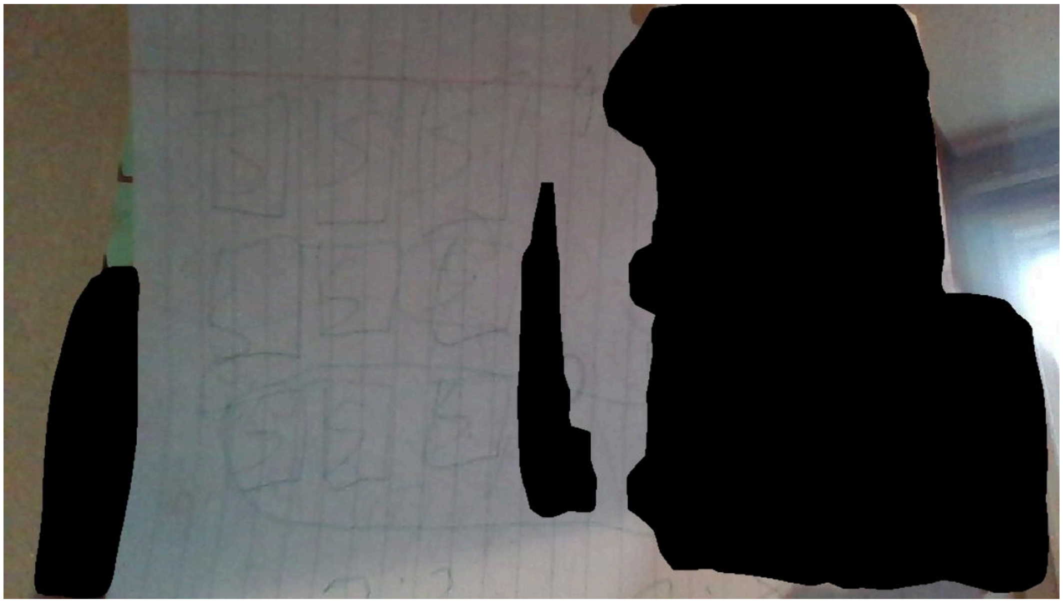

[redacted] [redacted] [redacted]

Figure 3.6.2.2: A student answer of problem ID 1622991 that is very unclear, confusing, and has considerable PII interference

For all three problems tested, a nearly identical or, at the very least, remarkably similar prompt template was used for the first prompt of each. The only aspect that differed between the first prompt of the three problems was the problem description. The problem description was specified in the first prompt of every problem in order to assist GPT with identifying properly the problem that it is supposed to assess with the image analysis tool. In some of the images submitted to GPT, there were parts of other unrelated questions visible, which could potentially confuse GPT and hinder its ability to focus on the desired problem.

3.6.3 Examining the Utility of the DALL-E 3 Image Generation Plugin

Given that GPT-4 has the DALL-E 3 image generation plugin built-in, it would be valuable to examine what it can do in terms of generating diagrams and graphs. If the plugin shows promising results in generating mathematical visualizations, they could potentially be used to supplement the textual explanations. A particular area of interest was to examine how well it would perform in generating an empty cartesian coordinate plane. If GPT-4 performs accurately and consistently on this task, it would warrant exploring further whether it would be capable of graphing linear functions and then more complex functions. The prompts provided to GPT in order to examine its ability to perform this task are as follows sequentially:

1. Can you generate an empty cartesian coordinate plane that a middle school student can use to draw equations?
2. can you try to lay it flat and take up the entire width and height of the image? Can you also make sure that the axes are labeled with valid and incremental numerical values?
3. Is there any other way you can make this without using DALLE?
4. Can you use any of these methods that you outlined above in order to generate the empty coordinate plane?
5. Can you plot a linear equation on that plane?
6. 10 + 5x

3.6.4 Examining the Utility of the Wolfram Plugin in GPT Plugins Mode

Discovery of the plugins mode and the Wolfram Plugin in GPT, which has been downloaded and installed from the Plugins store, indicated the potential to check the accuracy of GPT's calculations and solutions to math problems, particularly solutions to simple arithmetic and algebra questions. The prompts provided to GPT to explore the capabilities of the Wolfram plugin are as follows sequentially:

1. What can the plugins mode do?
2. How can you have access to two different plugins when I have only one plugin called "Wolfram" enabled to the best of my knowledge?
3. Can you guarantee that arithmetic calculations will be accurate? Can you provide examples to backup your assertion?
4. Did you utilize the plugin in order to check the accuracy of each of those results?
5. Yes can you show me an example of its usage?
6. What else is possible with this plugin? Can you give a comprehensive demo with multiple examples?

The prompts above were designed to be as broad as possible to allow GPT the ability to demonstrate all of the different ways that it is able to utilize the Wolfram plugin.

3.6.5 Examining the Efficacy of Using Markdown In GPT Prompts

The idea to prompt using markdown originated from the observation that GPT had a strong tendency to represent any tables or other figures using markdown syntax in its responses. The frequent use of markdown indicated with a strong likelihood that GPT had a relatively high level of proficiency, familiarity, and ease in utilizing markdown. Therefore, we decided to test out prompting on a problem involving a table by converting it to markdown syntax first instead of relying on the image analysis tool to extract information accurately from the image of the table. Problem ID 1057711 met this criteria. With the table being converted to markdown, the prompts provided to GPT are as follows sequentially:

1. Graph the following table.

```
| x  | y |
|----|---|
| 3  | 1 |
| 6  | 2 |
| 9  | 3 |
| 12 | 4 |
```

The above prompt simply attempted to test whether GPT would be able to provide a correct and understandable solution to the problem in markdown format or whether it would miss any of the information represented in the table.

3.6.6 Examining the Efficacy of Zero-Shot and One-Shot Prompting

Using the same problem from Section 3.6.5 (Problem ID 1057711), zero-shot prompting and one-shot prompting were tested. Zero-shot prompting involves giving GPT a problem that it has most likely not already been trained on and examining whether it can come up with a correct or acceptable solution without having seen any examples whatsoever of a correct solution. One-shot prompting relies on providing GPT with one student answer image with the correct solution in order to prime it or "train" it in the hopes that it performs better at identifying and evaluating the additional student answers that it receives in subsequent prompts for it to evaluate. The markdown syntax for representing the table in the prompts has been preserved from section 3.6.5. The prompts provided to GPT in order to test zero-shot prompting are as follows sequentially:

1. You are a stellar middle school math teacher that is dedicated to helping their students learn and understand math concepts to the best of their ability. Sometimes your students struggle with some math problems that you have assigned them. It is your responsibility to assess the work that they have submitted, identify any mistakes, and explain to them at the appropriate grade level an explanation to help them fix their mistake or clarify their understanding. You will be assessing student responses to the following math problem:

   Graph the following table.

x	y
3	1
6	2
9	3
12	4

   You will be provided with student answers in subsequent messages.

2.

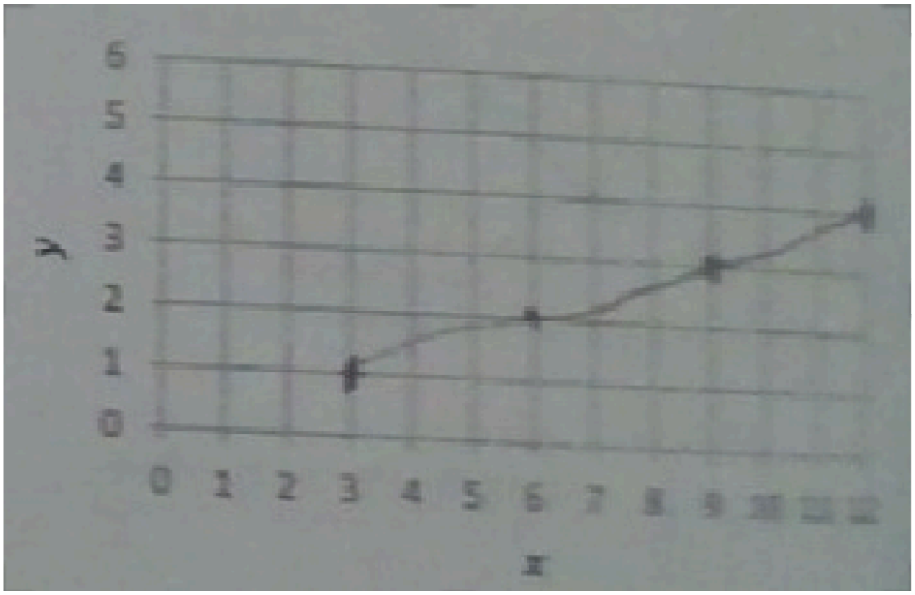


Figure 3.6.6.1: The first student answer provided to GPT for zero-shot prompting

3.

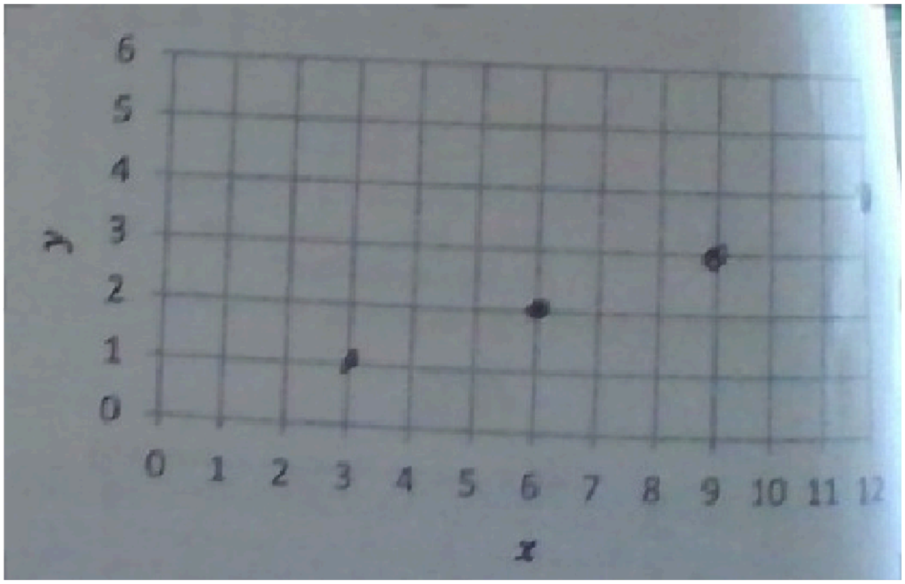


Figure 3.6.6.2: The second student answer provided to GPT for zero-shot prompting

4.

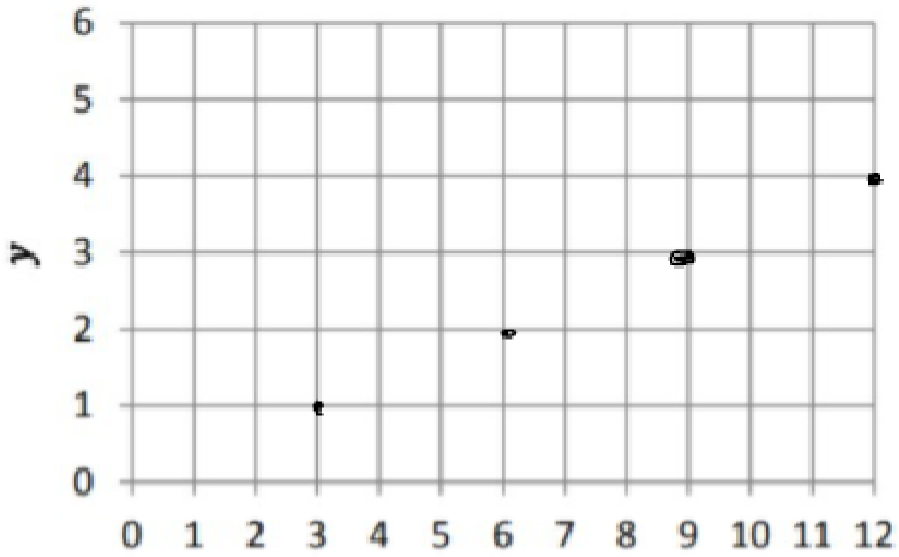


Figure 3.6.6.3: The third student answer provided to GPT for zero-shot prompting

A similar style of prompting was used for testing the one-shot prompting approach using the same problem that was used in the zero-shot prompting. The prompts that were provided to GPT are as follows sequentially:

1. You are a stellar middle school math teacher that is dedicated to helping their students learn and understand math concepts to the best of their ability. Sometimes your students struggle with some math problems that you have assigned them. It is your responsibility to assess the work that they have submitted, identify any mistakes, and explain to them at the appropriate grade level an explanation to help them fix their mistake or clarify their understanding. You will be assessing student responses to the following math problem:

   Graph the following table.

x	y
3	1
6	2
9	3
12	4

I have included an image of a sample correct answer as a point of reference if needed.

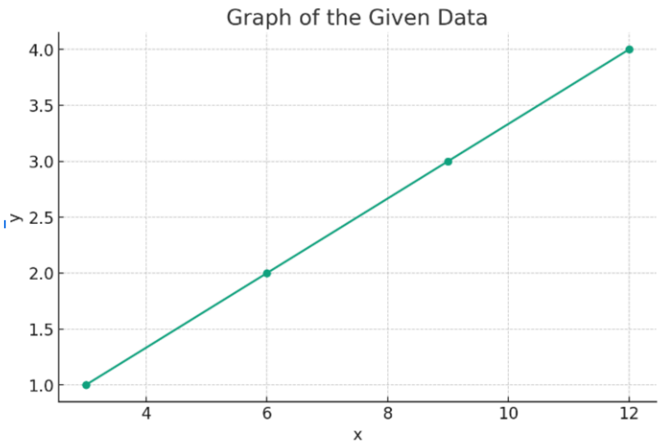


Figure 3.6.6.4: The correct answer example image provided to GPT for one-shot prompting

You will be provided with student answers in subsequent messages.

4. Results // Analysis

4.1 Overview

This section covers our analysis of our results for our five different projects: Image Cropper and PII Blocker, Image Clustering, Image Watermarking, Dataset Sharing, and GPT Prompting.

4.2 Image Cropper and PII Blocker

This section covers our results from our implementations of the Insert Image and Take Photo options in the ASSISTments website.

4.2.1 Insert Image

The “Insert Image” option performed its intended function effectively. After the user uploads images, they are able to crop the images they uploaded. This benefits both ASSISTments and the teachers: for ASSISTments, there is less PII in the image; for the teachers, there are fewer distractions and the work is clearer. Our final design is shown in Figure 4.2.1.1.

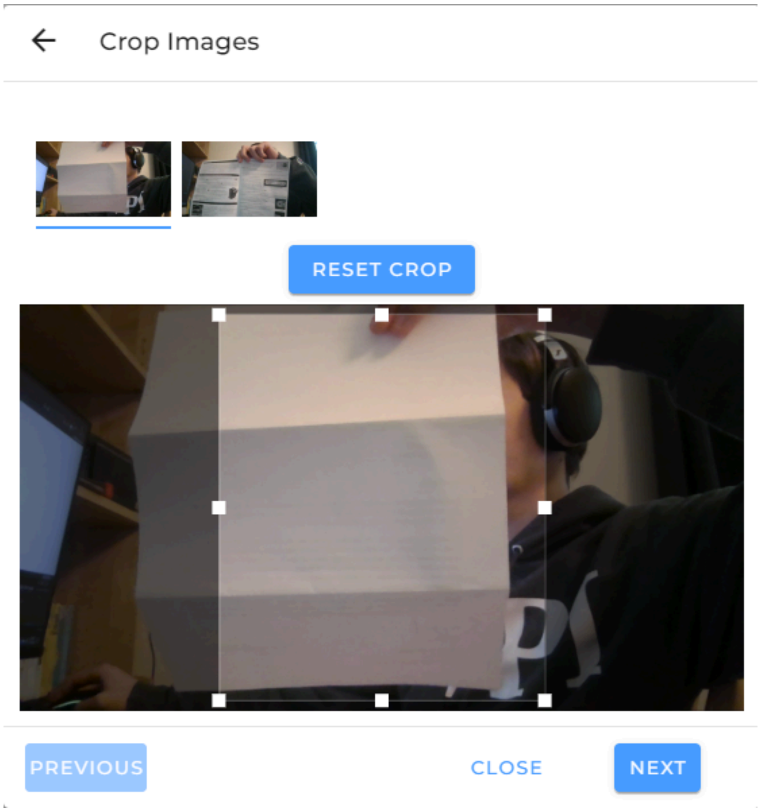


Figure 4.2.1.1: The final design of the Insert Image option.

4.2.2 Take Photo

The "Take Photo" option performed its intended function effectively. When an individual takes a photo, the image is cropped to only include the area within the bounding box. This means that if students properly place their work within the bounding box, this option will maximize the area of the final image that the student uploads as containing their work while minimizing most of the extraneous background material and PII. Our final design is shown in Figure 4.2.2.1.

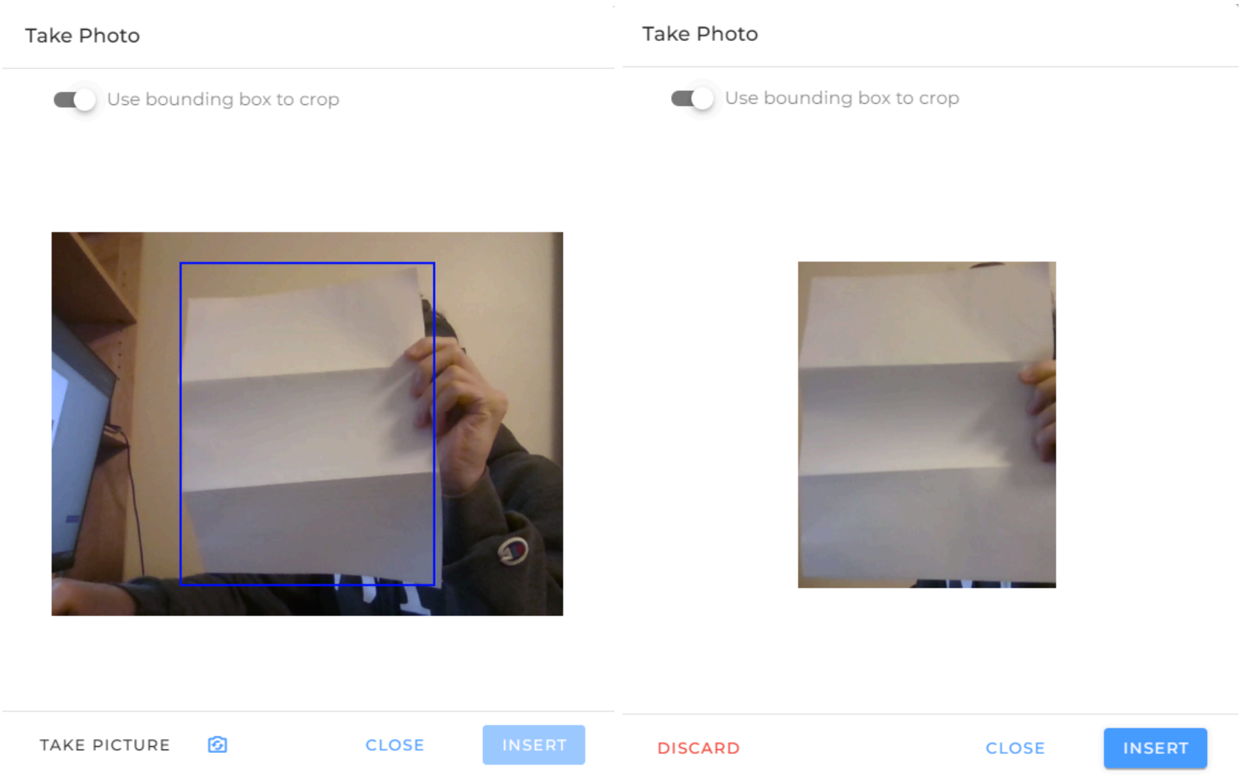


Figure 4.2.2.1: The Take Photo option when the user is lining up the paper (left), and after the user has taken the photo (right).

4.3 Image Clustering

This section covers our findings for the four different distance metrics we tested, along with the three different preprocessing effects. We will be using the scoring system discussed in Section 3.3.6, where we combine two problems into one large problem and attempt to separate the problems via clustering.

4.3.1 MSE

MSE as a distance metric failed to meaningfully differentiate between the two problems across all preprocessing effects (See Table 4.3.1.1). The slight differences in the image counts in each cluster can be attributed to random chance, and no pattern is shown. MSE also failed to properly cluster “Edge Images”, presumably because they share the same black background color, which makes it near-impossible for MSE to differentiate between them.

Table 4.3.1.1

Our results using MSE as a distance metric.

Cluster #	No Preprocessing		Cropping		Edge Images		Cropped Edge Images	
Counts	# P1	# P2	# P1	# P2	# P1	# P2	# P1	# P2
0	49	33	34	37	159	159	1	12
1	0	6	47	32	0	0	1	1
2	71	63	0	0	0	0	4	11
3	0	0	17	13	0	0	10	8
4	12	15	16	9	0	0	9	14
5	18	12	32	47	0	0	4	4
6	9	30	13	21	0	0	130	109

4.3.2 ORB

Our results using ORB as a distance metric were lackluster. As Table 4.3.2.1 shows, ORB failed to distinguish between the two problems in “No Preprocessing”, “Cropping”, and “Cropped Edge Images”. The “Edge Images” section shows some promising results: with the exceptions of Clusters 0 and 3, there is a definitive split between the Problem 1 and Problem 2 results; however, Cluster 3 contains over half of the images in the dataset, so the accuracy of ORB is still very limited.

Table 4.3.2.1

Our results using ORB as a distance metric.

Cluster #	No Preprocessing		Cropping		Edge Images		Cropped Edge Images	
Counts	# P1	# P2	# P1	# P2	# P1	# P2	# P1	# P2
0	68	52	16	8	24	14	45	35
1	0	0	71	85	7	2	20	8
2	2	0	9	4	12	2	21	47
3	41	52	58	60	62	109	53	33
4	37	48	1	1	22	6	6	22
5	10	7	0	1	17	0	14	13
6	1	0	4	0	5	26	0	1

4.3.3 Template Match

Using Template Match as a distance metric failed to produce suitable results for any combination of preprocessing effects. However, it was able to somewhat distinguish between Problem 1 and Problem 2. As Table 4.3.3.1 shows, there are no clusters with a similar amount of images from each problem, with the exception of “No Preprocessing”, Cluster 1, “Cropping”, Cluster 6, “Cropped Edge Images”, Cluster 3. The Template Match function provided by OpenCV does not account for rotation or scaling differences, which limits the amount of images it can correctly compare, so these results make sense.

Table 4.3.3.1

Our results using Template Match as a distance metric.

Cluster #	No Preprocessing	Cropping	Edge Images	Cropped Edge Images

Counts	# P1	# P2	# P1	# P2	# P1	# P2	# P1	# P2
0	3	2	0	1	20	7	82	120
1	72	93	4	7	44	26	6	0
2	4	1	10	0	3	1	27	2
3	13	7	6	25	2	0	13	16
4	26	16	3	6	79	116	24	0
5	28	13	11	17	9	9	1	2
6	13	27	125	103	2	0	6	19

4.3.4 CLIP Embeddings

As Table 4.3.4.1 shows, CLIP produces stellar results when “Cropping” was the preprocessing method. CLIP also produced suitable results with “No Preprocessing”, which is what will likely be used when detecting CWAs in real-world applications, since “Cropping” requires the images to be manually cropped. CLIP failed to meaningfully differentiate images when “Edge Images” preprocessing was used, presumably since CLIP was trained on real-world images, which are very different from the edge images. CLIP performed slightly better on “Cropped Edge Images”, although it was nowhere near the results from “No Preprocessing” and “Cropping”.

Table 4.3.4.1

Our results using the angle between CLIP embeddings as a distance metric.

Cluster #	No Preprocessing		Cropping		Edge Images		Cropped Edge Images	
Counts	# P1	# P2	# P1	# P2	# P1	# P2	# P1	# P2

0	15	4	2	68	38	37	12	4
1	7	0	0	62	1	11	9	6
2	14	6	2	24	72	52	4	75
3	119	45	12	0	5	7	7	4
4	0	73	36	1	25	44	30	2
5	4	19	45	2	4	7	78	65
6	0	12	62	2	14	1	19	3

4.4 Image Watermarking

This section covers our findings from the suite of robustness tests we ran on the three watermarkers, as well as the performance results when adding a "key" character to the encoded message. See Section 3.4.3 for a more detailed description of the robustness tests we ran.

4.4.1 Preliminary Testing

We found that invisible-watermark and blind_watermark passed three tests: compress, overlay, and mask (See Table 4.4.1.1). Both failed to decode the message when a mild or severe crop was applied to the image.

Table 4.4.1.1

Results from our robustness tests on invisible-watermark and blind_watermark. The images were encoded with the 32-character message "testaaaabbbbcccctestaaaabbbbcccc". The invisible-watermark image was encoded using the dwtDctSvd method.

Action	invisible-watermark	blind_watermark

Compress (size -46%)	Pass	Pass
Noise	Fail	Fail
Decrease Brightness (~50%)	Fail	Fail
Overlay (draw stuff on the image)	Pass	Pass
Mask (block out sections)	Pass	Pass
Mild crop (~60 px removed from right)	Fail	Fail
Severe crop (removed from all sides)	Fail	Fail
Resize (50% of original size)	Fail	Fail
Rotate (30 degrees)	Fail	Fail

4.4.2 Final Results

We found that our implementation was successfully able to pass the mild and severe crop tests (See Table 4.4.2.1). As a bonus, it also passed the noise test. This is a major improvement over the original invisible-watermark implementation.

Table 4.4.2.1

The results of our robustness tests on invisible-watermark, blind-watermark, and our own watermarker implementation. The images were encoded with the message "_thisisa32charactersamplemessage". Bolded results show where our implementation passed a test that both invisible-watermark and blind-watermark failed.

Action	invisible-watermark	blind_watermark	Our Implementation
Compress (size -46%)	Pass	Pass	Pass
Noise	Fail	Fail	**Pass**

Decrease Brightness (~50%)	Fail	Fail	Fail
Overlay	Pass	Pass	Pass
Mask (block out sections)	Pass	Pass	Pass
Mild crop (~60 px removed from right)	Fail	Fail	**Pass**
Severe crop (~60 px removed from all sides)	Fail	Fail	**Pass**
Resize (50% of original size)	Fail	Fail	Fail
Rotate (30 degrees)	Fail	Fail	Fail

4.4.3 Usage of a "Key" Character

Additionally, we found that the presence of a "key" character let us discard 310 outputs, reducing the number of garbage outputs from 319 to 9. This greatly reduces the time it takes to find the correct output.

4.4.4 Flaws and Limitations

The image watermarker we developed has four limitations:

1. Some encoded bits will not preserve correctly when the image is written to disk, leading to the decoded message being incorrect.
2. Images must be larger than 128x128.
3. When an image has not been cropped, the correct solution will always be the first element in the list that is returned, but this is not guaranteed to be true when an image is cropped. So the list of possible decoded messages will have to be manually reviewed to find the correct solution.

Because of issue 3, we recommend that anyone using this algorithm save the message that each image was encoded within a database, so the results produced from decoding can be cross-checked with the messages in the database. Additionally, because of issue 1, we recommend that all watermarks are cross-checked by either a human or some sort of distance algorithm (e.g. Levenshtein distance) in case one or two characters are decoded incorrectly.

Also, because of issue 1, we recommend that each image that is encoded is written to disk and then decoded before it is shared in case the image cannot be correctly decoded after it is written to disk.

4.5 Dataset Sharing

Our automated dataset sharing tool was successfully able to share datasets with all of our testers.

4.6 GPT Prompting

This section covers the responses generated by GPT addressing the prompts mentioned in sections 3.6.2 - 3.6.6 and providing an analysis of the most important findings from each response. This section will mostly include the principal findings or key takeaways from each response rather than the actual responses themselves for the sake of brevity. However, the actual responses are available in Appendix C for a more detailed review if needed.

4.6.1 Analysis of Responses to Initial Prompting Approach and Testing

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts regarding problem id 1754531:

1. **Principal Findings From Response:** GPT failed to recognize that two distinct triangles can be made from SAA (Side-Angle-Side) and ASA (Angle-Side-Angle) Theorems. See Appendix C1.1 for the full generated response.
2. **Principal Findings From Response:** GPT alerts to the low quality of the image, so it seems that a lot of the information it has included in this response most likely came from the previous context of this chat. See Appendix C1.2 for the full generated response.
3. **Principal Findings From Response:** GPT fails to recognize that the number 1 in 100° is hidden by the letter A that represents a vertex label of the triangle in the image. There are also slight changes that it made in what it believes is the correct answer and reasoning from the previous answer. See Appendix C1.3 for the full generated response.
4. **Principal Findings From Response:** GPT seems to misidentify the position of the angles shown in the image. It also seems to slightly change the reasoning it provides. See Appendix C1.4 for the full generated response.
5. **Principal Findings From Response:** GPT provides very general guidelines that the student should have in their solution but is not able to assess the student answer directly. See Appendix C1.5 for the full generated response.

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts regarding problem ID 1755170:

1. **Principal Findings From Response:** GPT seems to be potentially confused about which point on the graph is actually marked. It also seems to complain that the resolution is low which is potentially why the y-axis value cannot be read. See Appendix C2.1 for the full generated response.
2. **Principal Findings From Response:** GPT seems to be the most through and correct for this problem so far. However, it is still unable to identify the alignment of the marked point with the y-axis. See Appendix C2.2 for the full generated response.
3. **Principal Findings From Response:** GPT does a relatively correct and thorough assessment. However, it does complain about not being able to assess the marked point in comparison to the y-axis due to the low image resolution. See Appendix C2.3 for the full generated response.
4. **Principal Findings From Response:** This response corresponds to a prompt provided with a wrong student answer example to examine how GPT would assess such answers. The explanation provided in the response is confusing and contains mistakes. It made several attempts at guessing the y-intercept value used by the student. Additionally, it did not read the range of the y-axis correctly. See Appendix C2.4 for the full generated response.
5. **Principal Findings From Response:** This response corresponds to a prompt provided with a wrong student answer example to examine how GPT would assess such answers. The explanation in the response seems to get the correct increment value for the x-axis. It also makes suggestions about educating the student on using more common terminology than "max vertical intercept." The response seems to indicate that GPT might have

forgotten or confused the context of the original problem. See Appendix C2.5 for the full generated response.

6. **Principal Findings From Response:** This response corresponds to a prompt that is unrelated to the original question but is just to ensure that GPT still remembers the context of the original problem. The response provided indicated that GPT does remember the context. See Appendix C2.6 for the full generated response.
7. **Principal Finding From Response:** This response corresponds to a prompt provided with a blank / wrong student answer example to examine how GPT would assess such answers. The response correctly identifies that the image submitted by the student is blank. See Appendix C2.7 for the full generated response.

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts regarding problem ID 1622991:

1. **Principal Findings From Response:** For the most part, GPT provided a correct explanation except for the usage of the phrase "two sets of three circles" when it should have been the reverse, "three sets of two circles." See Appendix C3.1 for the full generated response.
2. **Principal Findings From Response:** GPT tries to make a distinction between maple syrup and seeds in the diagram that the student has drawn. However, it is slightly perplexing how it is so sure when the quality of the image is fairly low, and it is difficult to make that distinction even by a human looking at the image. See Appendix C3.2 for the full generated response.

4.6.2 Analysis of Responses to Testing GPT's Ability to Generate Graphs

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts regarding testing GPT's ability to generate graphs:

1. **Principal Findings From Response:** GPT seems to have attempted to generate a cartesian coordinate plane utilizing the DALL-E 3 internal plugin. However, the lines are distorted, the axes labels and values do not make any sense, and the graph is 3-dimensional and lopsided. See Appendix C4.1 for the full generated response.
2. **Principal Findings From Response:** GPT still makes a 3-dimensional and lopsided graph, but changed the colors and style of the graph. It was unsuccessful at generating a flat image of a cartesian coordinate plane. See Appendix C4.2 for the full generated response.
3. **Principal Findings From Response:** When asked if there is another way to create this cartesian coordinate plane without using DALL-E, GPT proceeded to list various general ways such as drawing it on paper, using Excel, or using python libraries such as matplotlib. However, it does not actually produce the cartesian plane itself, instead relying on the reader to execute the actions or procedure that it has advised. See Appendix C4.3 for the full generated response.
4. **Principal Findings From Response:** When asked to use one of the methods that it had outlined from the previous response, GPT resorted to writing some python code and utilizing the Advanced Data Analysis internal plugin to generate the graph as part of the output of running the python code. See Appendix C4.4 for the full generated response.
5. **Principal Findings From Response:** This response is not very eventful as GPT is only inquiring about what linear equation to plot in response to asking it to plot a linear

equation for demonstration purposes. It was presumed that GPT would just produce an example linear equation of its own, but that turned out not to be the case. See Appendix C4.5 for the full generated response.

6. **Principal Findings From Response:** GPT successfully utilized the Advanced Data Analysis plugin in order to produce an acceptable graph for the equation $y = 10 + 5x$. However, the y-axis increment labels seem to be very squished together while there is ample white space on the graph for that to not occur. See Appendix C4.6 for the full generated response.

4.6.3 Analysis of Responses to Testing the Wolfram Plugin

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts exploring the capabilities of the Wolfram plugin:

1. **Principal Findings From Response:** GPT gave an overview of the plugins mode by stating that it was a way for GPT to interact with third party services and external sources to fetch some specific information. It proceeded to give a slight overview of the inner workings and capabilities of the Wolfram Plugin which was the only one installed and activated at the time. It revealed the fact that internally the Wolfram plugin had two more specialized plugins that composed it: the Wolfram Alpha plugin and the Wolfram Cloud Plugin. The Wolfram Alpha plugin specializes in performing computations and querying information from Wolfram Alpha's knowledge base in math, science, history, geography, and various other fields. This is probably the one that will be utilized the most for our purposes. The Wolfram Cloud plugin allows it to execute Wolfram Language code for

more custom queries and data analysis not available via one of the pre-built Wolfram queries. See Appendix C5.1 for the full generated response.

2. **Principal Findings From Response:** This response corresponds to a clarifying question about how GPT had access to two different plugins under one Wolfram plugin since that assertion sounded like it was potentially a mistake. It responded by clarifying again that it had access to both the Wolfram Alpha and Wolfram Cloud plugins. See Appendix C5.2 for the full generated response.
3. **Principal Findings From Response:** This response corresponds to asking GPT whether it can ensure that all arithmetic calculations will be accurate and to show some examples of performing correct computations. While GPT did provide some examples of it performing the correct computations, it did not appear that it was utilizing the Wolfram plugin. See Appendix C5.3 for the full generated response.
4. **Principal Findings From Response:** This response corresponds to asking GPT whether it had utilized the Wolfram plugin in order to verify the accuracy of the computations it performed in the previous response. GPT responded by stating that it had not. See Appendix C5.4 for the full generated response.
5. **Principal Findings From Response:** This response corresponds to simply asking GPT again to utilize the Wolfram Plugin and show some example usages. It performed a simple addition problem and seems to have also generated an image of the number line representation of said problem. See Appendix C5.5 for the full generated response.
6. **Principal Findings From Response:** This response corresponds to inquiring about the full capabilities of the Wolfram plugin and providing a comprehensive demo with multiple examples. For one of the problems, it was able to perform the calculation and

generate the graph of the quadratic curve as well as the number line representation of the solution. The rest of the demonstrations that it had outlined had been halted due to a network error pertaining to making more requests to the plugin. See Appendix C5.6 for the full generated response.

4.6.4 Analysis of Responses to Prompts Utilizing Markdown

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts testing the utilization of markdown:

1. **Principal Findings From Response:** This response corresponds to converting the image of the math problem, specifically a table containing some points that need to be plotted, into a markdown representation in hopes that it would facilitate GPT's accuracy in comparison to relying solely on the image analysis plugin. In the response, GPT seems to have utilized the data analysis plugin from the very beginning in order to generate the correct graphical answer which reduces the potential error that comes from relying solely on GPT's native capabilities in solving graphing questions. This prompt seems to have inadvertently tested GPT's ability to write and execute python code to produce the graph of the equation and solve the problem rather than testing its native ability to approach the problem. See Appendix C6.1 for the full generated response.

4.6.5 Analysis of Responses to Zero-shot and One-shot Prompting

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts testing zero-shot prompting:

1. **Principal Findings From Response:** The first response corresponds to a prompt introducing GPT to the problem description that it will be working with and letting it know that it will receive images of student answers in subsequent prompts. In the response, GPT confirmed that it understood the task and was waiting for student responses. See Appendix C7.1 for the full generated response.
2. **Principal Findings From Response:** GPT gets the x-axis scale incrementation wrong and fails to identify that the axes are labeled. It accurately identifies the lack of smoothness or straightness of the line drawn by the student. It does correctly identify that this student answer is correct, but how it is certain of this assessment while getting some aspects of the student answer wrong is not entirely clear. See Appendix C7.2 for the full generated response.
3. **Principal Findings From Response:** GPT correctly identifies that the points are plotted correctly. It also accurately identifies the scale used and the axes labels. It also correctly identifies that the points are not connected. GPT marks the student correct conditionally based on the type of the data used, either discrete or continuous. GPT essentially argues that if the nature of the data is discrete, then the points should not be connected. However, if the nature of the data is continuous, then the student should draw a line connecting the points plotted. However, it seems that this dilemma that GPT is facing is beyond the scope of the simple graphing skills that this problem attempts to assess in students. See Appendix C7.3 for the full generated response.
4. **Principal Findings From Response:** GPT correctly identifies that the student answer is correct, the points are plotted correctly, identifies the scale of both axes incrementing by 1 unit correctly, and identifies the corresponding axes labels correctly. However, it still

continues to question the discrete vs. continuous nature of the data, which arguably detracts from the simple graphing skills that the problem is assessing in students. See Appendix C7.4 for the full generated response.

These are the principal findings or key takeaways from each of the responses generated by GPT to the prompts testing one-shot prompting.

1. **Principal Findings From Response:** This response corresponds to a prompt that introduces GPT to the problem description and one example of a correct answer in order to prime GPT on what it should look for when assessing other student answers. In the response, GPT confirms that it has understood the task and is waiting for student answers to assess in subsequent prompts. See Appendix C8.1 for the full generated response.
2. **Principal Findings From Response:** GPT complains that the image provided is unclear and is unsure about what the image specifically contains. However, GPT does provide decent feedback to the student describing what criteria they need to have in order to be considered correct. See Appendix C8.2 for the full generated response.

5. Conclusions & Recommendations

This section covers our findings from our five projects, as well as our recommendations for future work.

5.1 Image Cropper and PII Blocker

Our implementation of the Insert Image button within the ASSISTments workbench allows students to crop images after they are uploaded, which greatly reduces the amount of PII in the images.

Our Take Photo modifications give students a box to suggest where they should hold the paper when taking a picture. After the picture is taken, students are given the option to crop the photo to just the paper, which, like the Insert Image button, greatly reduces the amount of PII in the images.

5.2 Image Clustering

We published our results for Template Match and CLIP embeddings in a paper submitted to LAK24, an educational conference. The paper pitted our results against those of hierarchical clustering. The full paper can be found in Appendix A.

We recommend using the angle between CLIP embeddings as a distance metric for images since it outperformed the three other distance metrics in all categories. In terms of preprocessing effects, we recommend using the “Cropping” preprocessing effect since that produced the best results by far.

5.3 Image Watermarking

We found that both invisible-watermark and blind_watermark were unable to decode an image after it had been cropped, so we do not recommend using either library to create crop-resistant watermarks.

Our modified implementation of invisible-watermark was successfully able to decode an image after it had been cropped, while simultaneously passing the same robustness tests invisible-watermark and blind_watermark passed. For this reason, we recommend using our modified implementation of invisible-watermark to generate and decode crop-resistant invisible watermarks.

5.4 Dataset Sharing

We found that Google Apps Script, along with a Python backend server, was successfully able to fully automate the process of sharing the 28k Images Dataset .

5.5 GPT Prompting

We found that while GPT-4 has made significant advancements in its mathematical reasoning capabilities as well as its image analysis capabilities in order to extract considerable amounts of information from images of student answers, it still seems that it is not as reliable, consistent, or correct as we would like, specially with problems involving any visuals, diagrams, and graphs. There is still a lot of work to do in terms of improving GPT-4 to be more consistent and accurate. We would recommend using this tool with a fair amount of caution and skepticism.

Although extensions of GPT via plugins such as Wolfram seem very promising, other components such as the image analysis tool are much less reliable and inconsistent.

5.6 Future Work

This subsection covers the limitations of our work, along with any future work we propose.

5.6.1 Image Clustering

CLIP embeddings perform best on images with the "Cropped" preprocessing effect, but currently the images need to be manually cropped. Potential solutions to this problem include:

1. Fine-tuning a CLIP model to perform better on uncropped images
2. Using an existing paper-detection model to crop the image to just the paper
3. Fine-tuning an existing object detection model (such as YOLOv7) to crop each image to just the relevant content
4. Using the hierarchical clustering method discussed in the paper in Appendix A, which performed significantly better than the k-means clustering we used.

5.6.2 Image Watermarking

There is one addressable limitation in the image watermark: the constraint that the message must be exactly 32 characters long. We recommend that in the future, research is done to address this issue.

5.6.3 GPT Prompting

There are a lot more tests that can be done in order to explore the full extent to which GPT can analyze images of student math work. One possible endeavor for the future would be to explore more comprehensively zero-shot, one-shot, and few-shot prompting. In our tests, we were only able to test a very limited number of examples that did not clearly make evident whether zero-shot or few-shot sampling actually made a noticeable difference in how GPT analyzed the image. Another potential endeavor to undertake in the future would be to fine-tune GPT specifically on a large number of middle school math problems in order to produce a more specialized and accurate model that performs better in assessing student math work. Although the internal structure of GPT is closed-source due to OpenAI's policies, it would also be helpful to potentially explore the possibility of fine-tuning the image analysis feature somehow so that GPT can become better at detecting and extracting relevant information from these images more consistently and accurately.

References

Allotey, C. (2023, January 26). Options API vs composition API. Options API vs Composition API. https://vueschool.io/articles/vuejs-tutorials/options-api-vs-composition-api/

ASSISTments. (n.d.-a). *About Us*. ASSISTments. https://new.assistments.org/about

ASSISTments. (n.d.-b). *Quick comments*. ASSISTments. https://new.assistments.org/individual-resource/quick-comments

E-Trials. (2023, March 7). *Mathnet competition*. MathNet Competition. https://www.etrialstestbed.org/projects/mathnet-competition

Papers With Code. (n.d.). *ImageNet*. Papers With Code. https://paperswithcode.com/dataset/imagenet

Pinto, J. (n.d.). Assistments announces university winners of Mathnet Competition: Assistments blog. ASSISTments Announces University Winners of MathNet Competition. https://new.assistments.org/blog-posts/assistments-announces-university-winners-of-mathnet-competition

Stanford University. (n.d.). *About ImageNet*. ImageNet. https://www.image-net.org/about.php

Appendices

Appendix A

Image Clustering Paper

This appendix contains the full paper submitted to LAK24 containing our work on image clustering. This paper was submitted for anonymous peer-review, so all identifying information has been blinded.

Do These Students Have Similar Strategies? Clustering Math Work in Uploaded Images on an Online Learning Platform

BLINDED AUTHOR, Blinded Institute, blinded
BLINDED AUTHOR 2, Blinded Institute, blinded
BLINDED AUTHOR 3, Blinded Institute, blinded
BLINDED AUTHOR 4, Blinded Institute, blinded
BLINDED AUTHOR 5, Blinded Institute, blinded
BLINDED AUTHOR 6, Blinded Institute, blinded
BLINDED AUTHOR 7, Blinded Institute, blinded

This exploratory study delves into the complex challenge of analyzing and interpreting student responses to mathematical problems, typically conveyed through image formats within online learning platforms. The main goal of this research is to identify and differentiate various student strategies within a dataset comprising image-based mathematical work. A comprehensive approach is implemented, including various image representation, preprocessing, and clustering techniques, each evaluated to fulfill the study's objectives. The exploration spans several methods for enhanced image representation, extending from conventional pixel-based approaches to the innovative deployment of CLIP embeddings. Given the prevalent noise and variability in our dataset, an ablation study is conducted to meticulously evaluate the impact of various preprocessing steps, assessing their potency in eradicating extraneous backgrounds and noise to more precisely isolate relevant mathematical content. Two clustering approaches—k-means and hierarchical clustering—are employed to categorize images based on student strategies that underlies their responses. Preliminary results underscore the hierarchical clustering method could distinguish between student strategies effectively. Our study lays down a robust framework for characterizing and understanding student strategies in online mathematics problem-solving, paving the way for future research into scalable and precise analytical methodologies while introducing a novel open-source image dataset for the learning analytics research community.

CCS Concepts: • **Computing methodologies** → **Image processing**; **Image representations**; **Cluster analysis**.

Additional Key Words and Phrases: Open-ended questions, Image responses, Embeddings, Clustering, Mathematics

ACM Reference Format:
Blinded Author, Blinded Author 2, Blinded Author 3, Blinded Author 4, Blinded Author 5, Blinded Author 6, and Blinded Author 7. 2018. Do These Students Have Similar Strategies? Clustering Math Work in Uploaded Images on an Online Learning Platform. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 15 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In recent years, online learning platforms have witnessed substantial growth, accelerated by factors such as globalization, advancements in technology, and more recently, global challenges like the COVID-19 pandemic [30]. This transition to

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

digital platforms has led to an unprecedented influx of diverse student data, including mathematics education. Among the various data types about student learning captured through this mathematics education platform, image-based submissions — capturing handwritten equations, sketches, and diagrams — are particularly noteworthy.

Such image submissions, often termed as 'visual artifacts' of learning, provide an unparalleled window into students' thought processes, their conceptual understanding, and their problem-solving strategies [25]. They transcend the limitations of traditional text-based responses, enabling educators to decipher nuances like hesitation in strokes, the sequence of problem-solving, or even errors and corrections made during the process [3, 27]. This level of granularity can be pivotal in understanding not just the 'what' but the 'how' students are learning, allowing educators to provide precise feedback and tailored instruction based on students' strategies and reasoning reflected in their responses. Moreover, analyzing images can aid in automatically identifying common misconceptions, patterns of thought, and even predicting potential hurdles a student might face in the future. For instance, the way a student sketches a parabola or labels a geometric figure might give hints about their comprehension of underlying concepts [8]. Such insights can be instrumental in the timely remediation of learning gaps and fostering a more supportive and efficient mathematics learning environment.

However, the richness and complexity of these image-based submissions also pose distinctive challenges in their analysis and interpretation. Traditional analytic techniques, designed primarily for textual or numeric data, fall short when applied to images, necessitating the development of innovative methods attuned to the nuances of visual data [1]. Some pioneering efforts have been made to analyze hand-drawn diagrams or sketches using image recognition techniques to provide instant feedback in domains like engineering and physics [8].

In the context of mathematics, earlier studies have often relied on simplistic pattern recognition methods to classify hand-written equations and geometrical sketches [26]. Nevertheless, the diverse nature of student strategies, especially when conveyed through images, calls for a more holistic and nuanced approach. It is this gap in the literature that our study seeks to address, integrating advanced embedding techniques and sophisticated clustering algorithms to delve deeper into the world of image-based student responses to evaluate students' underlying strategies. Our study embarks on a mission to decipher image-based student submissions by addressing three pivotal research questions:

1. **RQ1**- Does the incorporation of embeddings enhance our capacity to differentiate between distinct categories of students' mathematical reasoning and strategies depicted in images?
2. **RQ2**- How does the choice of preprocessing method impact the differentiation process?
3. **RQ3**- To what degree does the utilization of different clustering techniques enhance our ability to distinguish between various students' responses?

In navigating these questions, we aim to enhance the empirical evidence through detailed, step-by-step comparisons, evaluating whether students' response strategies and reasoning in mathematical problems can be assessed using a sample of image data derived from two sample math problems. Recognizing the scarcity of specialized datasets in this domain, we are releasing our meticulously curated, image-based dataset to the broader research community. This open-source resource will not only serve as a valuable foundation for further studies in the realm of image-based learning analytics but also stimulate the development of novel analytical methods specifically tailored for such data.

2 RELATED WORKS

2.1 Online Learning Platforms in Math Education

The digital transition in education has witnessed the rise of online platforms explicitly tailored for various subjects, with mathematics being a prominent area of focus. This shift towards online math platforms has been catalyzed by the increasing need for flexible, accessible, and interactive learning environments [15, 16]. The digital transition in education has ushered in the rise of online platforms, explicitly tailored for various subjects, mathematics being a notably prominent focus. This shift towards online math platforms has been catalyzed by an ever-increasing need for flexible, accessible, and interactive learning environments [15, 16].

The COVID-19 pandemic further expedited the transition to online learning, leading to an augmented application of online learning platforms in K-12 mathematics classrooms [32]. Online mathematics learning platforms provide various advantages, making mathematics learning more accessible and personalized. These platforms enable personalized and self-paced learning, facilitating students' engagement with mathematical concepts and practices at their convenience [14]. They also provide interactive learning and assessment resources, which cater to students' individual needs and aid in establishing more effective and efficient learning environments. Moreover, these platforms allow for potentially instant feedback and progress monitoring through online assessments. The adoption of automated grading of student responses [4, 6, 23], analysis of students' writing patterns and discourse [2], and generation of teacher feedback [12, 17], have been rigorously demonstrated in the previous literature.

2.2 Automated Scoring in Online Math Assessment

Automated scoring systems in mathematics education have predominately focused on evaluating students' computational skills [11], problem-solving strategies [5], and, occasionally, the procedural steps undertaken during problem-solving [29]. The recent introduction of transformer-based models into the scoring systems remarkably extends their capabilities, particularly in improving scoring accuracy [4, 35], ensuring scoring consistency and fairness [10], and expanding these models for the generation of timely feedback [6]. Despite such advancements, one major focus area remains the incorporation of new response formats, such as image-based responses.

2.2.1 Automated Scoring of Image-based Math Responses. Image-based responses require students to create a visual representation of their work using a traditional paper and pencil approach or using digital media and upload their work to online learning platforms. GeoGebra[13] and Desmos[9] are some examples of computer-based applications that allow students to interact with graphs and algebraic expressions. While these kind of tools and support for visual representation of answers exists in online learning platforms, some teacher still prefer the traditional approach of paper and pencil and some others use a blend of both in their classrooms.

While previous automated assessment methods have relied heavily on text-based constructed responses, where students type an answer directly into the online learning platforms [4, 11, 35], recent works have explored a diverging type of student responses, particularly, image-based responses. Baral et al.[3] proposed methods to auto-score open-ended mathematics questions containing text and image responses [3]. Using optical character recognition and deep learning models like CLIP reduced scoring errors for mixed text and image responses over models that only handled text. As online assessments expand the types of responses they allow, automated scoring techniques must evolve to handle multimedia response formats.

2.3 Approaches in Image Processing and Analysis

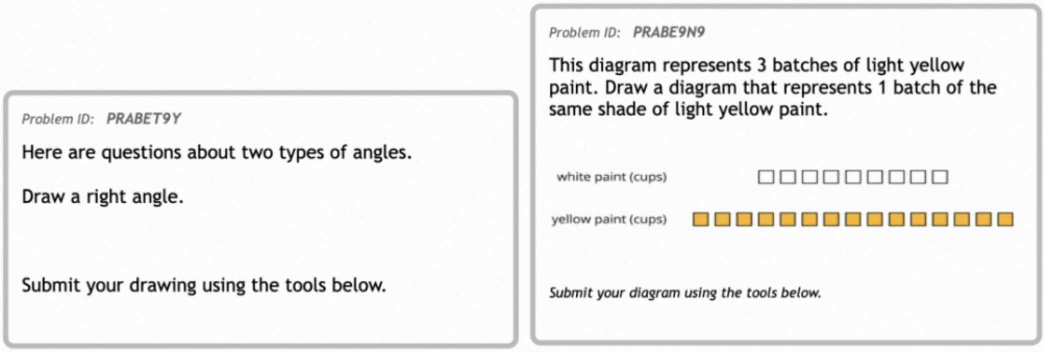
2.3.1 Embeddings for Image Representation. One of the primary roles of embeddings in image analysis is to facilitate the understanding and representation of visual content within them. In recent years, image embeddings have emerged as a transformative technique in image processing and analysis, offering powerful ways to represent and understand visual content. The features used in image analysis are often of high dimension, thus necessitating techniques like feature extraction for handling multi-modal features for image classification tasks. [3, 24]. Luo et al. [24] have explored the domain of multi-modal multi-task feature extraction, highlighting the advantages of leveraging multiple modalities in such scenarios.

The CLIP (Contrastive Language-Image Pretraining) model, introduced by Radford et al. [31], is an image classification model based on transformer architecture, commonly used in natural language processing tasks. CLIP learns joint image-text embeddings, allowing images to be encoded into a shared space with natural language descriptions. This versatile representation has found applications in various image analysis tasks, from image classification to zero-shot learning. In our analyses, we explore the use of image embeddings from a pre-trained CLIP model as a method of image representation and further investigate the ability of such a pre-trained model in the domain of mathematics education.

3 DATA COLLECTION AND PARTICIPANTS

In this study, we utilize a dataset of student responses to open-ended mathematics questions taken from an online learning platform. The main goal of this study is to analyze and compare various clustering methods with image processing techniques to identify and distinguish various student approaches to solving a math problem within the student-uploaded works. As such we mainly look into the response from students that are images. This dataset was collected from a BLINDED Online learning platform, from a middle school mathematics classroom. The students were assigned mathematics assignments using the BLINDED learning platform; which consisted of both close-ended and open-ended problems. For open-ended problems, the students were allowed to provide either a textual response or they had the option to write their answer and upload the image of their work directly to the learning platform.

The dataset, in addition to the image responses from students, consists of a numeric assessment score given by a teacher. The scores for these responses are on an ordinal 5-point scale ranging from 0 to 4.



Problem ID: ***PRABET9Y***

Here are questions about two types of angles.

Draw a right angle.

Submit your drawing using the tools below.

Problem ID: ***PRABE9N9***

This diagram represents 3 batches of light yellow paint. Draw a diagram that represents 1 batch of the same shade of light yellow paint.

Submit your diagram using the tools below.

Fig. 1. The two open-response mathematics problems used for the analyses in this study, Problem 1 (on the left), and Problem 2 (on the right)

For this study, we selected two specific mathematics problems which had mostly image-based responses from students. Figure 1 shows these two math problems. The first problem which we call "Problem 1" throughout this study is a 4th-grade problem based on the Relationship of Angles, while the second problem "Problem 2" is a 6th-grade math based on Defining Equivalent Ratios.

Problem 1, had 159 scored image responses, while Problem 2 had about 269 scored images in total. For Problem 2 we randomly sampled 159 images to balance out the dataset for the purpose of the study. The final dataset includes 318 image-based responses from 318 unique students who answered the 2 math problems. This dataset of images was scored by 51 different teachers.

Utilizing this dataset [1] , we perform an exploratory analysis through the application of various clustering techniques in order to distinguish different approaches taken by students for solving math problems. We discuss the methods taken in detail in the following sections.

4 METHODS

4.1 OpenCV Template Match vs CLIP

In this subsection of the methods, we perform a comparison of different image representation techniques to identify the best approach for differentiating and grouping various students' math works. We compare the raw pixel matching using OpenCV's template match techniques, with context-rich encoding of images from a popular deep-learning method called "CLIP" for clustering images.

4.1.1 OpenCV Template Match. OpenCV, a popular computer vision library, offers a technique known as template matching, for comparing different images. We leverage this method to identify similarities in our dataset of images and group these using K-means clustering. Template matching is typically used for finding instances of a template image (a small image) within a larger target image. The goal of this is to find regions in the target image that closely match the template. In our case, we adapt this technique to compare and group images by using a bidirectional approach. We use the "TM_CCORR_NORMED" method for template matching. This method calculates the cross-correlation between the images with the highest value indicating the best match.

Initially, we apply template matching by matching Image A onto Image B. This means we treat Image A as the template and try to find the best matches in Image B. Next, we reverse the process and match Image B onto Image A. Now, Image B serves as the template, and we look for matches in Image A. After performing template matching in both directions, we obtain match scores for Image A matched onto Image B and Image B matched onto Image A. To determine the overall similarity between the two images, we consider the minimum of these match scores. This is done to account for cases where one image may match well with the other, but the reverse might not be true. For example, if image A is a perfectly drawn right-angled triangle and image 2 is a plain sheet of graph paper then image A matches on to image B well but not vice versa. We perform this for each of the images in our dataset. The resulting match scores for pairs of images used as a distance metric are then utilized as input for k-means clustering, a common method for grouping similar data points.

4.1.2 CLIP Embeddings. CLIP (Contrastive Language-Image Pre-training)[31] is an image classification model based on transformer architecture that offers a versatile and context-rich means for representing visual content. Introduced

---

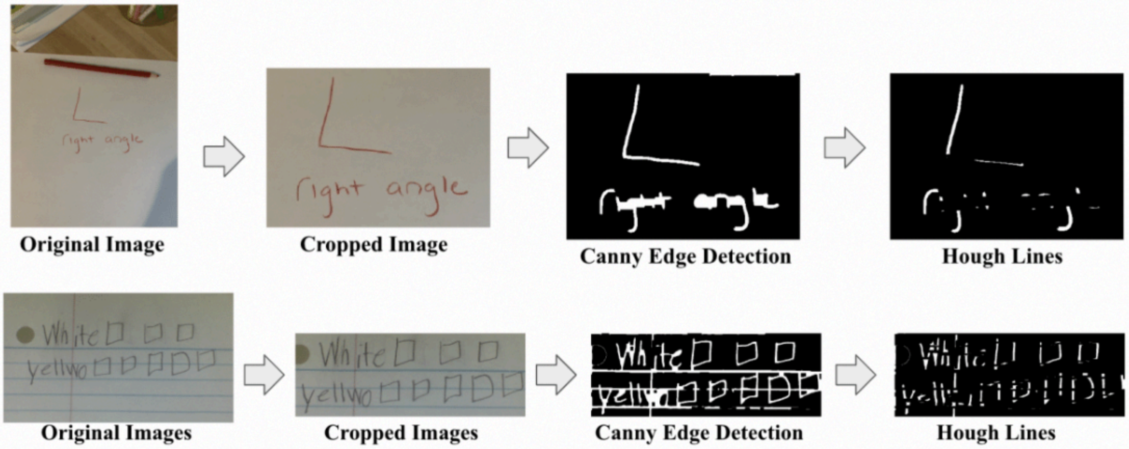
[1] A curated dataset of these images with cropped background is shared through this URL: https://osf.io/9a8xq/?view_only=f0cd8d45acfd49f3be8aa1f0c1eb375b. For review purposes, we only share the cropped image data, but a more comprehensive dataset will be shared for the final version.

by OpenAI, this model harnesses the power of a vision-language transformer architecture and is able to encode both natural languages (text) and images in the same vector space by using a multi-modal pre-training approach. While the CLIP model was initially designed for the combination of text and images, its embeddings can be effectively used for image representation tasks independently of textual information.

In this study we use the *"clip-vit-large-patch14-336"* version of the CLIP model, to generate an embedding vector representation for each of the images of students' math works. These image embeddings encapsulate both the visual characteristics and semantic content of the images, effectively allowing us to understand not only what the images contain but also what they mean making them potential for various image-related tasks. To assess the similarity between images encoded by CLIP, we use an angle-based metric. For this particular method, instead of relying on the traditional Euclidean distance, we measure the angle between the CLIP embedding vectors and further use these as input to the K-means clustering. The number of clusters is chosen based on the elbow plot for the accuracy scores with cluster size k ranging from 2 to 12.

4.2 Ablation Study on the Impact of Preprocessing

The original dataset of images posed a challenge due to the presence of background noise, including elements like students' faces, backgrounds, and other non-relevant content alongside the mathematical content. In addition to this, there are differences in the images aside from the mathematical content and background, coming from the use of different types of papers (like graph, math, or plain paper, the use of digital media vs. conventional pencil and paper, etc. As such, the image representations may pick up on the non-relevant content, tampering with the results of clustering. For this reason, we apply the following preprocessing steps and conduct an ablation study of these different methods to identify the best-suited preprocessing method for the image clustering tasks.



Original Image
Cropped Image
Canny Edge Detection
Hough Lines

Original Images
Cropped Images
Canny Edge Detection
Hough Lines

Fig. 2. Example image response from Problem 1(top) and Problem 2(bottom) with the applied preprocessing steps.

4.2.1 Image cropping. In this step, a meticulous inspection was performed on each image within the dataset, to identify the relevant math content. The objective was to crop and isolate the core math content within the images while removing any extraneous noise or background elements. This step not only enhanced the clarity of the images but also facilitated more precise template matching and CLIP-based analysis.

4.2.2 Edge Detection. To further refine image preprocessing, Canny edge detection [7] was employed. This technique identifies prominent edges within the images. By emphasizing edges and contours, this method enhances the ability to detect and differentiate key mathematical elements within the images.

4.2.3 Hough Lines. The Probabilistic Hough lines algorithm [21] was used in conjunction with the Canny edge detection to address the specific challenges related to graph and math paper lines within the images. While Canny edge detection effectively identifies edges, it may also detect lines originating from the underlying graph paper or grid, which are unrelated to the mathematical content. The Hough lines algorithm is used to identify and remove these extraneous lines if they occur repeatedly and are parallel to each other.

The example of student responses in the dataset with applied preprocessing steps are shown in Figure 2.

4.3 K-means vs Hierarchical Clustering

To unravel latent structures and patterns, which are associated with students' mathematical response strategies, within the dataset, we conducted two clustering techniques, including K-means [22] and hierarchical clustering [28] as our primary analytical approaches. Euclidean distance was chosen as the distance metric to measure the similarity between CLIP embeddings.

Hierarchical clustering allowed us to construct a dendrogram that organized the image responses into a hierarchy of clusters, with each node representing a group of similar responses. Determining the optimal number of clusters is crucial for obtaining meaningful and interpretable clustering results. We addressed this challenge by employing silhouette analysis, a widely-used technique for evaluating the quality of clustering. By systematically varying the number of clusters from 2 to 10 based on the baseline threshold (t=2), which were identified from the dendrogram, and computing the silhouette score for each configuration, we identified the optimal number of clusters that maximized the cohesion within clusters and separation between them. This process ensured that our clustering was both statistically robust and reflective of the underlying patterns in students' problem-solving approaches.

4.4 Evaluation Metrics

We used three commonly adopted cluster evaluation metrics: the Gini score (or Gini index) [18], the purity score [33], and the silhouette score. The Gini score, also known as the Gini index or Gini coefficient, quantifies the inequality or impurity within a cluster. Often applied in hierarchical clustering or decision tree algorithms, it measures how mixed or heterogeneous the elements within a cluster are. In clustering analysis, a Gini score of 0 indicates perfect purity, signifying all elements in the cluster belong to the same class or category—in our case, the same mathematical strategy or reasoning to solve a problem.

The purity score is another metric used to evaluate clustering quality, especially in unsupervised learning and clustering algorithms. It gauges how closely the elements within a cluster relate to the same class or category. In clustering analysis, a purity score of 1 signifies perfect purity, indicating all elements in the cluster pertain to a singular class or category. A diminished purity score implies the elements within the cluster are diverse and may affiliate with multiple classes or categories.

However, both the Gini and purity scores can be susceptible to the effects of increasing cluster sizes, potentially leading to skewed evaluations. To counteract this limitation, we also employed the silhouette score. The silhouette score measures an object's similarity to its own cluster in contrast to other clusters. Its values lie between -1 and 1. A high silhouette score indicates the object aligns well with its own cluster and poorly with neighboring clusters. Conversely, a low silhouette score suggests potential misclustering. If most objects boast high silhouette scores, the clustering configuration is deemed appropriate. Yet, if many objects present low or negative scores, the clustering may encompass too many or too few clusters.

5 RESULTS

Our final results indicate that both K-means and Hierarchical clustering analyses performed comparably in identifying and clustering images based on their underlying mathematical reasoning. A total of seven clusters were retrieved from K-means clustering, and based on the distance threshold of 2 according to the ward linkage, Hierarchical clustering with the cluster size of 4 to 11 were compared for further evaluation.

5.1 OpenCV Template Match vs. CLIP

Comparing the GINI index and the purity score between OpenCV Template Match and CLIP embedding yielded distinct results as shown in Table 1. The CLIP embedding, when combined with K-means clustering, consistently showcased an enhancement in the clustering outcomes, marked by a lower GINI index and a higher purity score.

5.2 Ablation Study on the Impact of Preprocessing

The most optimal GINI index was obtained using the cropped image, registering at 0.157, paired with a notably high purity score of 0.894 as seen in Table 1. The original image also achieved an equivalent purity score of 0.894. The influence of other image processing techniques seemed marginal in augmenting the clustering outcomes. Specifically, the images processed with edge and hough line techniques recorded the peak GINI index of 0.396 and the lowest purity score of 0.676. Based on these findings, we juxtaposed the clustering methodologies, namely K-means and Hierarchical clustering, using the original and cropped images. This comparison was intended to further assess improvements in image clustering predicated on students' mathematical reasoning.

Table 1. Clustering Evaluation Metrics

Processing and Clustering Methods	Cluster size	GINI index	Purity score
OpenCV Template Match			
Original image	7	0.254	0.787
CLIP and K-means Clustering			
Original image	7	0.169	0.896
Cropping	7	**0.157**	**0.894**
Cropping and Edge	7	0.203	0.864
Edge and Hough Lines	7	0.386	0.676
Cropping, Edge, and Hough Lines	7	0.297	0.789
CLIP and Hierarchical Clustering			
Original image	10	0.11	0.917
Cropping	10	**0.088**	**0.943**
Cropping and Edge	10	0.221	0.837
Edge and Hough Lines	10	0.238	0.831
Cropping, Edge, and Hough Lines	10	0.227	0.839

5.3 K-means vs. Hierarchical Clustering

Hierarchical clustering was analyzed across a range of cluster sizes from 4 to 11. This range was determined based on a threshold of 2 when inspecting the dendrogram using the ward linkage method (see Figure 3). The results highlighted that a cluster size of 4 yielded the highest silhouette score for both original and cropped images. On the other hand, a cluster size of 10 resulted in the lowest GINI index and the highest purity score. It's important to note that the silhouette score for cluster size 10 was notably better than for cluster size 11. To further scrutinize the clustering methods' efficiency in unveiling students' intrinsic mathematical reasoning depicted in their image responses, additional comparisons between the K-means and hierarchical clustering outcomes were undertaken. The final clustering analyses were performed comparing the *cropped* images clustered into 7 groups using K-means, against the total of 10 clusters derived from hierarchical clustering.

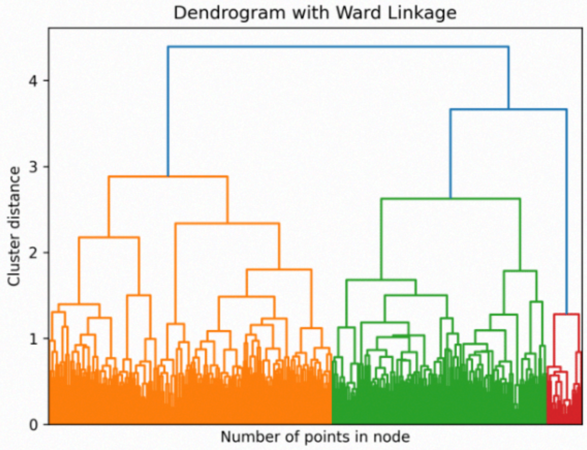


Fig. 3. Dendrogram for the hierarchical cluster with ward linkage method to identify the baseline threshold

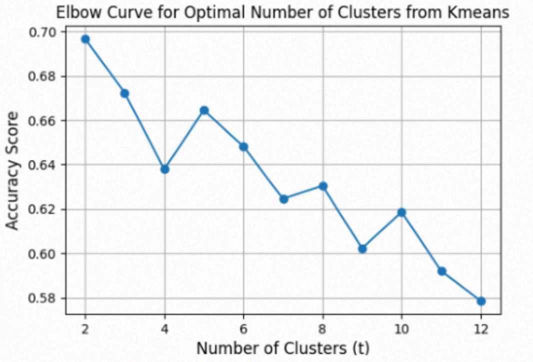


Fig. 4. Elbow plot with accuracy scores of the k-means clusters

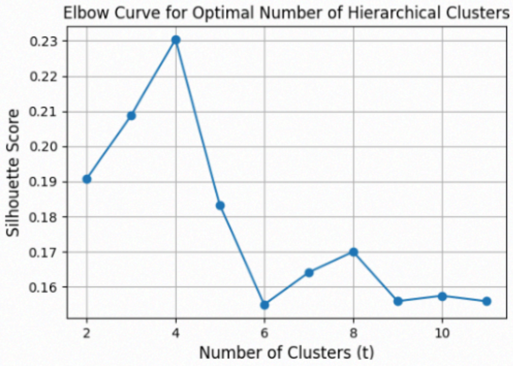


Fig. 5. Elbow plot of silhouette scores of the hierarchical clusters

Table 2. CLIP and Hierarchical Clustering Results with the Ward linkage method

	Original Image			**Cropped Image**		
Cluster size	*GINI index*	*Purity score*	*Silhouette score*	*GINI index*	*Purity score*	*Silhouette score*
4	0.179	0.876	**0.230**	0.152	0.895	**0.205**
5	0.144	0.901	0.183	0.138	0.908	0.190
6	0.153	0.898	0.155	0.116	0.922	0.175
7	0.131	0.913	0.164	0.100	0.933	0.167
8	0.138	0.896	0.170	0.105	0.931	0.170
9	0.123	0.907	0.160	0.096	0.937	0.165
10	0.110	0.917	0.157	0.088	0.943	0.142
11	**0.100**	**0.924**	0.156	**0.087**	**0.944**	0.128

Table 3. Distribution of Image Responses across Clusters

Cluster	**Total Images**	**Problem 1**	**Problem 2**
K-means with Cropped images (t=7)			
3	56	0	**56**
2	58	1	**57**
4	18	1	**17**
7	51	**51**	0
6	50	**47**	3
5	23	**16**	7
1	62	**43**	19
Hierarchical Clustering with Cropped images (t=10)			
1	33	0	**33**
2	38	0	**38**
3	39	1	**38**
7	38	2	**36**
4	10	**10**	0
8	23	**23**	0
10	35	**35**	0
6	37	**36**	1
5	44	**38**	6
9	21	**14**	7

5.3.1 Clustering Performance Accuracy. The clusters acquired from the two methods of clustering with cropped images were compared based on their accuracy to retrieve the underlying students' mathematical reasoning represented in their image responses. Table 3 provides the final performance accuracy of the two clustering methods, identified by the distribution of the image responses that originated from the two math problems (i.e., Math Problem 1, Math Problem 2) across the clusters. These two math problems as shown in Figure 1 required students to approach and solve the problems with two distinctive mathematical reasoning.

The results indicate that both K-means and hierarchical clustering methods are able to differentiate the two distinct mathematical reasoning coming from the dataset of two different math problems as seen by the proportions of the images in each of the resulting clusters. Overall, both methods could clearly separated the math problems based into different cluster categories – such as Clusters 2, 3, 4, 6, and 7 in K-means and Clusters 1, 2, 3, 4, 6, 7, 8, and 10 in hierarchical clustering – with few exceptions.

In terms of the clusters that failed to clearly separate the two method problems, in K-means, out of the 7 resulting clusters, 2 clusters (Clusters 1 and 5) represented a considerable mix of image responses from both problems 1 and 2. Cluster 1 and 5 both had 70% of images from Problem 1 and 30% images from Problem 2. Similarly, for the hierarchical clustering, out of the 10 clusters, two of the clusters (Cluster 5 and 9) represented a mix of the image responses from both Problems 1 and 2. Overall, the performance accuracy to separate the responses based on the math problems, indicate that our hierarchical clustering approach could show slightly improved clustering results compared to the K-means with less clusters with a mix of image responses from the different problem categories.

5.3.2 Illustrative Examples from Hierarchical Clustering Results. We further qualitatively assessed the clusters to understand the characteristics of the clusters from the hierarchical clustering results. First, we evaluated the clusters (e.g., Clusters 4, 10, 3 and 7) that demonstrated clear separation between the two problems. Figure 6 presents some examples from the clusters 4, 10, 3 and 7, identified through hierarchical clustering using the cropped images. Clusters 4 and 10, present responses from Problem 1 in the dataset. All the responses in Cluster 4 "right angle" text written as a label to the drawn right angle. While cluster 10, groups the perfectly drawn right angles with "90°" marking as the label. Clusters 3 and 7, present responses solely from Problem 2. Cluster 3 picks up mostly on the textual format of responses, whereas Cluster 7 presents digital images which are the screenshots of the question with markings for the answer.

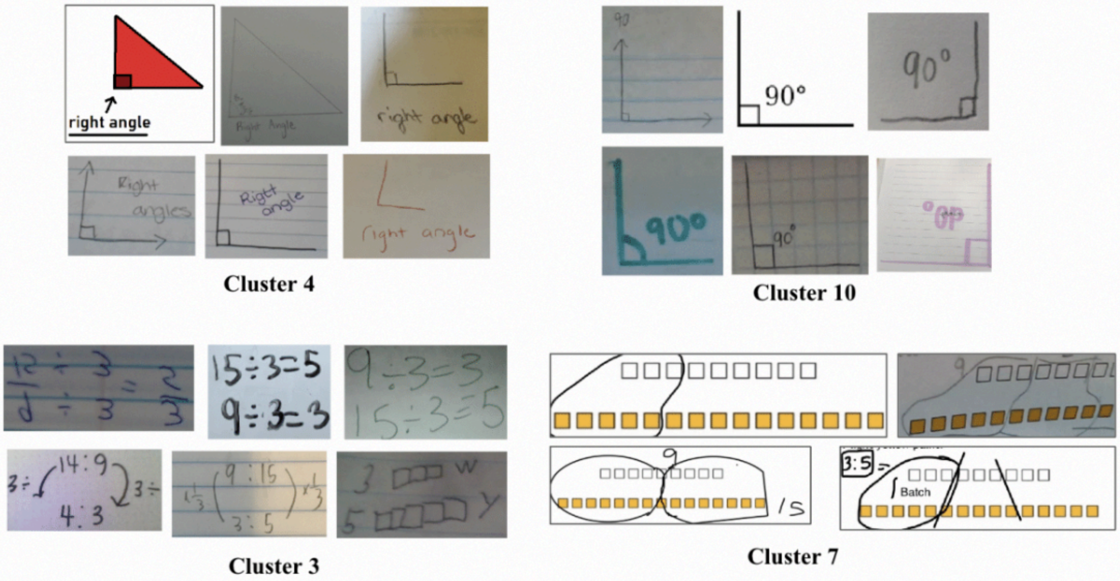
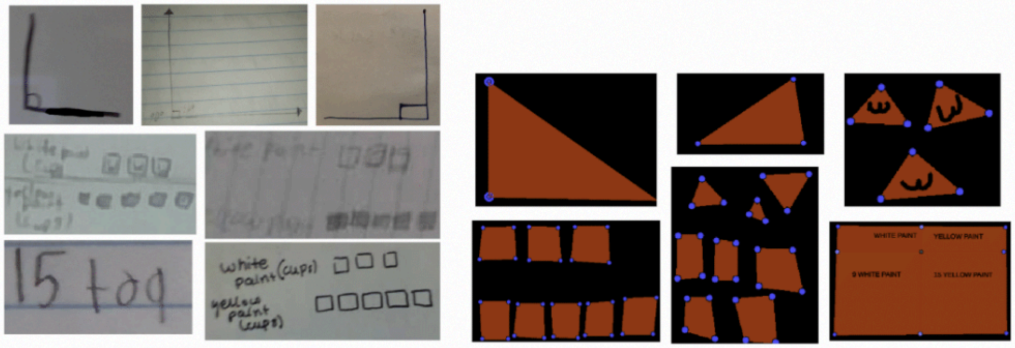


Fig. 6. Example images from the clusters with clear separations (Clusters 4, 10, 3, & 7) from the Hierarchical clustering method with cropped images.

Second, we evaluated the clusters (e.g., Clusters 5, 9) that showed a less clear separation of students' approaches seen in the images. The clusters presented a mix of images from both math problems grouped together into the same clusters. Figure 7 displays selected images from clusters 5 and 9, which were identified through hierarchical clustering. Cluster 5 predominantly consisted of images from Problem 1 (86%) with a minor portion from Problem 2 (13%). All images in this cluster were of handwritten math work on paper. A notable similarity among these images was the type of paper



Cluster 5 **Cluster 9**

Fig. 7. Example images from the clusters with *less* clear separations (Clusters 5 & 9) from the Hierarchical clustering method with cropped images.

used, as illustrated in Figure 7. On the other hand, Cluster 9 had a more balanced distribution with 66% of images from Problem 1 and 33% from Problem 2. Images in this cluster primarily represented digital submissions from both problems. The study's dataset featured images with a distinctive black background, adorned with shapes like triangles and squares filled in brown. Cluster 9 captured these characteristic images from both problems, as can be seen in Figure 7.

6 DISCUSSION

This research presents and compares different methods of image representation methods, preprocessing steps, and clustering techniques to identify and distinguish different types of student approaches seen in image-based responses. This study is a preliminary analysis conducted using a small dataset of 318 image responses from two mathematics problems, that sheds light on the application of unsupervised methods to distinguish different student approaches seen in image-based responses. The results from this exploratory analysis presented valuable insights into the use of suitable image representation methods along with preprocessing and clustering techniques for the analyses of image-based responses in mathematics.

The findings from the analyses indicate that CLIP embeddings provide a powerful means of representing and analyzing image-based student responses. Further, the conducted ablation study on the impact of the preprocessing step suggested that removing background noise and irrelevant features with the cropping of images enhanced the accuracy of both the clustering methods. However, it is noteworthy that the other two preprocessing steps – edge detection and the subsequent application of the Hough lines algorithm, had detrimental effects on the clustering outcomes. Edge detection method, while valuable for simplifying the representation of images, in the context of mathematical image responses seemed to oversimplify the data representation, potentially discarding some of the crucial information within these images. Moreover, the Hough lines algorithm, intended to remove extraneous grid lines, might have inadvertently interfered with the interpretation of certain mathematical components within the images. The Hough lines algorithm can be especially useful for finding prominent linear features, such as grid lines coming from graph paper. In our context, this method seems to capture and remove the horizontal line drawing in most of the right-angled triangle problems, as seen in the example in Figure 2. Further, it also tampered with some of the text in the images as seen in Figure 2.

In terms of two different clustering methods – K-means and Hierarchical – hierarchical clustering yielded better results in distinguishing various student approaches, potentially due to the varying factors addressed in previous literature [19, 20]. First, the robustness to outliers in the hierarchical clustering approach may have helped improve performance accuracy in our context. The presence of outliers can heavily affect centroid selection in K-means clustering approaches, where the varying sizes, shapes, and noisy features that frequently appear in image data could have had a negative effect. Second, hierarchical clustering can generally accommodate more flexible dissimilarity and similarity comparison measures through the linkage method, which can aid in constructing clusters that are flexible in both size and shape. This is particularly advantageous in cases where distance measures are computed from a large dimension of embedding, where the computation of similarity or dissimilarity becomes a less clear, and challenging problem [34].

7 LIMITATIONS AND FUTURE WORK

Despite the promising findings, our study encounters several limitations. Primarily, the dataset used for this analysis was relatively small, comprising 318 responses based on only two mathematics problems. This preliminary analysis aimed to compare various methods of image representation and clustering, seeking to identify the optimal unsupervised method. Expanding this dataset to include a broader variety of student responses in mathematics would enhance the generalizability of our approach.

Moreover, the dataset exhibited an uneven distribution of scores, with over 80% of the image responses receiving a full score of 4 from teachers. Scores, serving as labels, can be pivotal in analyzing the ability of image clustering methods to distinguish between correct and incorrect responses. Nevertheless, a dataset with an uneven distribution of correct versus incorrect responses presents a challenge in developing and evaluating these methods in a nuanced manner. Furthermore, the study utilized a manual cropping process for the images. Future work could explore developing automated methods for background removal.

Beyond the constraints and limitations inherent in the methodologies and techniques explored, analyzing image-based student work presents its own set of formidable challenges. These stem from the inherent variations and complexity found in the dataset of image-based responses. Factors such as variability in writing styles, different types of handwriting, the use of various symbolic notations, and the unstructured format of the responses all contribute to the complexity of these analysis methods.Additionally, the limited availability of a comprehensive dataset in the educational domain poses a significant challenge in constructing improved methods of analysis and support for these images.

8 CONCLUSION

In conclusion, this research offers valuable insights into the differentiation of student approaches in image-based responses. By leveraging advanced techniques of image representation, optimizing preprocessing steps, and conducting systematic analyses employing image clustering, we have taken significant steps toward this goal. Future work will continue to refine and expand these methods to further enhance educational outcomes and facilitate personalized learning experiences in the online education landscape.

ACKNOWLEDGMENTS

BLINDED FOR REVIEW

REFERENCES

[1] Jessica Andrews-Todd, Jonathan Steinberg, Michael Flor, and Carolyn M Forsyth. 2022. Exploring automated classification approaches to advance the assessment of collaborative problem solving skills. *Journal of Intelligence* 10, 3 (2022), 39.

[2] Michelle Banawan, Jinnie Shin, Renu Balyan, Walter L Leite, and Danielle S McNamara. 2022. Math Discourse Linguistic Components (Cohesive Cues within a Math Discussion Board Discourse). In *Proceedings of the Ninth ACM Conference on Learning@ Scale*. 389–394.

[3] Sami Baral, Anthony Botelho, Abhishek Santhanam, Ashish Gurung, Li Cheng, and Neil Heffernan. 2023. Auto-scoring Student Responses with Images in Mathematics. In *The Proceedings of the 16th International Conference on Educational Data Mining.*

[4] Sami Baral, Anthony F Botelho, John A Erickson, Priyanka Benachamardi, and Neil T Heffernan. 2021. Improving Automated Scoring of Student Open Responses in Mathematics. *International Educational Data Mining Society* (2021).

[5] Randy Elliot Bennett. 2011. Automated scoring of constructed-response literacy and mathematics items. *Retrieved April* 14 (2011), 2011.

[6] Anthony Botelho, Sami Baral, John A. Erickson, Priyanka Benachamardi, and Neil T. Heffernan. 2023. Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. *Journal of Computer Assisted Learning* 39, 3 (2023), 823 – 840.

[7] John Canny. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698.

[8] Melanie M Cooper, Mike Stieff, and Dane DeSutter. 2017. Sketching the invisible to predict the visible: From drawing to modeling in chemistry. *Topics in cognitive science* 9, 4 (2017), 902–920.

[9] David Ebert. 2014. Graphing projects with Desmos. *The Mathematics Teacher* 108, 5 (2014), 388–391.

[10] John A Erickson and Anthony Botelho. 2021. Is it fair? Automated open response grading. In *International Conference on Educational Data Mining.*

[11] John A Erickson, Anthony F Botelho, Steven McAteer, Ashvini Varatharaj, and Neil T Heffernan. 2020. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. 615–624.

[12] Joshua Grossman, Zhiyuan Lin, Hao Sheng, Johnny Tian-Zheng Wei, Joseph J Williams, and Sharad Goel. 2019. MathBot: Transforming online resources for learning math into conversational interactions. *AAAI 2019 Story-Enabled Intelligence* (2019).

[13] Markus Hohenwarter and Markus Hohenwarter. 2002. GeoGebra. *Available on-line at http://www. geogebra. org/cms/en* (2002).

[14] Mahmood H Hussein, Siew Hock Ow, Monther M Elaish, and Erik O Jensen. 2022. Digital game-based learning in K-12 mathematics education: a systematic literature review. *Education and Information Technologies* (2022), 1–33.

[15] Gwo-Jen Hwang, Sheng-Yuan Wang, and Chiu-Lin Lai. 2021. Effects of a social regulation-based online learning framework on students' learning achievements and behaviors in mathematics. *Computers & Education* 160 (2021), 104031.

[16] Muhammad Irfan, Betty Kusumaningrum, Yuyun Yulia, and Sri Adi Widodo. 2020. Challenges during the pandemic: use of e-learning in mathematics learning in higher education. *Infinity Journal* 9, 2 (2020), 147–158.

[17] Jennifer Jacobs, Karla Scornavacco, Charis Harty, Abhijit Suresh, Vivian Lai, and Tamara Sumner. 2022. Promoting rich discussions in mathematics classrooms: Using personalized, automated feedback to support reflection and instructional change. *Teaching and Teacher Education* 112 (2022), 103631.

[18] Lan Jiang, Huidong Chen, Luca Pinello, and Guo-Cheng Yuan. 2016. GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome biology* 17, 1 (2016), 1–13.

[19] B Karthikeyan, Dipu Jo George, G Manikandan, and Tony Thomas. 2020. A comparative study on k-means clustering and agglomerative hierarchical clustering. *International Journal of Emerging Trends in Engineering Research* 8, 5 (2020).

[20] Manju Kaushik and Bhawana Mathur. 2014. Comparative study of K-means and hierarchical clustering techniques. *Int. J. Softw. Hardw. Res. Eng* 2, 6 (2014), 93–98.

[21] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. 1991. A probabilistic Hough transform. *Pattern recognition* 24, 4 (1991), 303–316.

[22] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern recognition* 36, 2 (2003), 451–461.

[23] Nava L Livne, Oren E Livne, and Charles A Wight. 2007. Can automated scoring surpass hand grading of students' constructed responses and error patterns in mathematics. *MERLOT Journal of Online Learning and Teaching* 3, 3 (2007), 295–306.

[24] Yong Luo, Yonggang Wen, Dacheng Tao, Jie Gui, and Chao Xu. 2015. Large margin multi-modal multi-task feature extraction for image classification. *IEEE Transactions on Image Processing* 25, 1 (2015), 414–427.

[25] Peter Maclaren. 2014. The new chalkboard: the role of digital pen technologies in tertiary mathematics teaching. *Teaching Mathematics and Its Applications: International Journal of the IMA* 33, 1 (2014), 16–26.

[26] Mahshad Mahdavi, Richard Zanibbi, Harold Mouchere, Christian Viard-Gaudin, and Utpal Garain. 2019. ICDAR 2019 CROHME+ TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1533–1538.

[27] Filip Moons, Ellen Vandervieren, and Jozef Colpaert. 2022. Atomic, reusable feedback: a semi-automated solution for assessing handwritten tasks? A crossover experiment with mathematics teachers. *Computers and Education Open* 3 (2022), 100086.

[28] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378* (2011).

[29] Ryosuke NAKAMOTO, Brendan Flanagan, Taisei Yamauchi, Dai Yilling, Kyosuke Takami, and Horoaki Ogata. 2023. Enhancing Automated Scoring of Math Self-Explanation Quality using LLM-Generated Datasets: A Semi-Supervised Approach. (2023).

[30] Sumitra Pokhrel and Roshan Chhetri. 2021. A literature review on impact of COVID-19 pandemic on teaching and learning. *Higher education for the future* 8, 1 (2021), 133–141.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[32] İpek SARALAR-ARAS. 2022. Countries' responses to Covid-19 pandemic in K-12 education: Toward a digital education in mathematics. *Uluslararası Sosyal Bilimler Eğitimi Dergisi* 8, 2 (2022), 450–478.

[33] Satya Chaitanya Sripada and M Sreenivasa Rao. 2011. Comparison of purity and entropy of k-means clustering and fuzzy c means clustering. *Indian journal of computer science and engineering* 2, 3 (2011), 343–346.

[34] Michael Steinbach, Levent Ertöz, and Vipin Kumar. 2004. The challenges of clustering high dimensional data. In *New directions in statistical physics: econophysics, bioinformatics, and pattern recognition*. Springer, 273–309.

[35] Mengxue Zhang, Sami Baral, Neil Heffernan, and Andrew Lan. 2022. Automatic Short Math Answer Grading via In-context Meta-learning. *arXiv preprint arXiv:2205.15219* (2022).

Received 2 October 2023

Appendix B

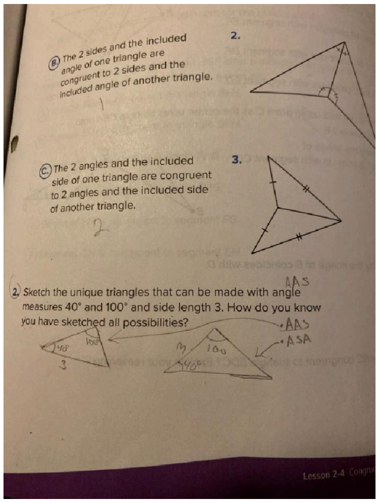
The Images in the GPT Prompts

This appendix contains the image attached to the first prompt and the second prompt onwards for problem IDs 1754531 and 1755170 from Section 3.6.2 Initial Prompting Approach and Testing.

Appendix B1. The Image Attached to the First Prompt and the Second Prompt Onwards for Problem ID 1754531.

This appendix contains the image attached to the first GPT prompt and the second prompt onwards for problem ID 1754531.

1. (Continuation of Prompt #1)



Ⓑ The 2 sides and the included angle of one triangle are congruent to 2 sides and the included angle of another triangle.

1

2.

Ⓒ The 2 angles and the included side of one triangle are congruent to 2 angles and the included side of another triangle.

2

3.

2. Sketch the unique triangles that can be made with angle measures 40° and 100° and side length 3. How do you know you have sketched all possibilities?

AAS

•AAS

•ASA

Lesson 2-4 Congru

Figure B1.1: The first student work image provided to GPT for Problem ID 1754531.

2. Here is another student response to the same question.

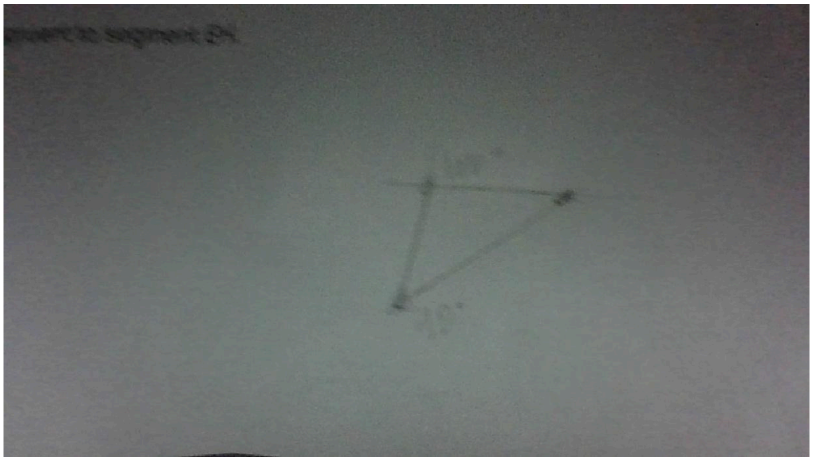


Figure B1.2: The second student work image provided to GPT for Problem ID 1754531.

3. Here is another student response

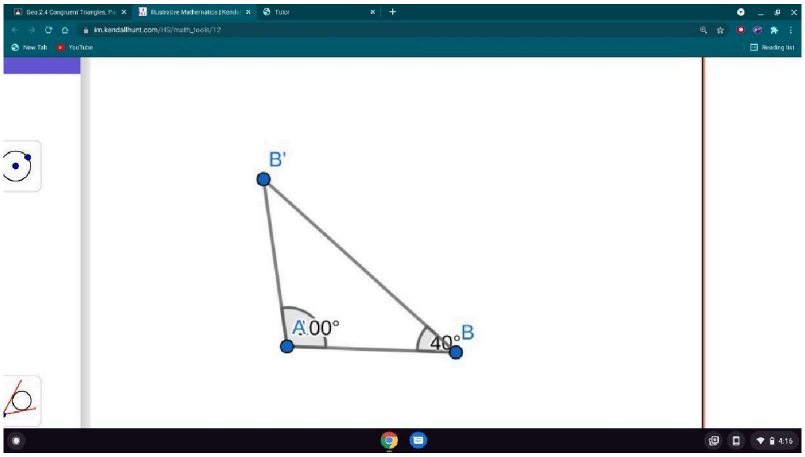
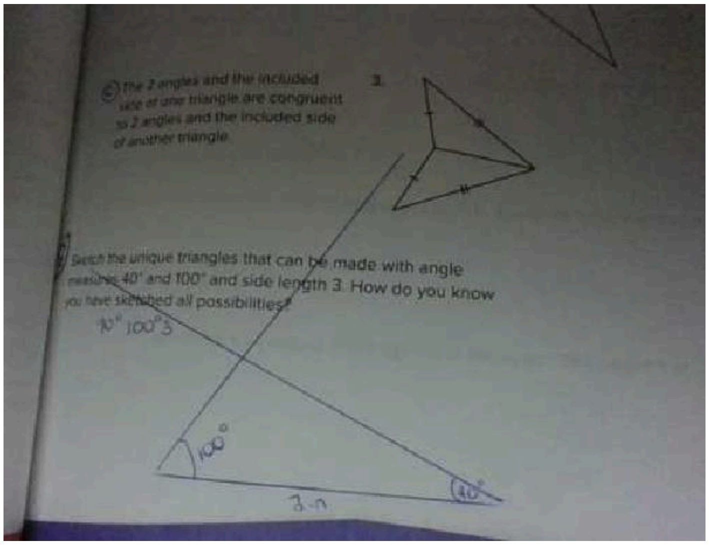

Geo.2.4 Congruent Triangles, Pa
Illustrative Mathematics | Kendall
Tutor
im.kendallhunt.com/HS/math_tools/12
New Tab
YouTube
Reading list
B'
A'
00°
40°
B
4:16

Figure B1.3: The third student work image provided to GPT for Problem ID 1754531.

4. Here is another student response



(c) The 2 angles and the included side of one triangle are congruent to 2 angles and the included side of another triangle.

3.

Sketch the unique triangles that can be made with angle measures 40° and 100° and side length 3. How do you know you have sketched all possibilities?

40° 100° 3
100°
40°
3 in

Figure B1.4: The fourth student work image provided to GPT for Problem ID 1754531.

5. Here is another student response

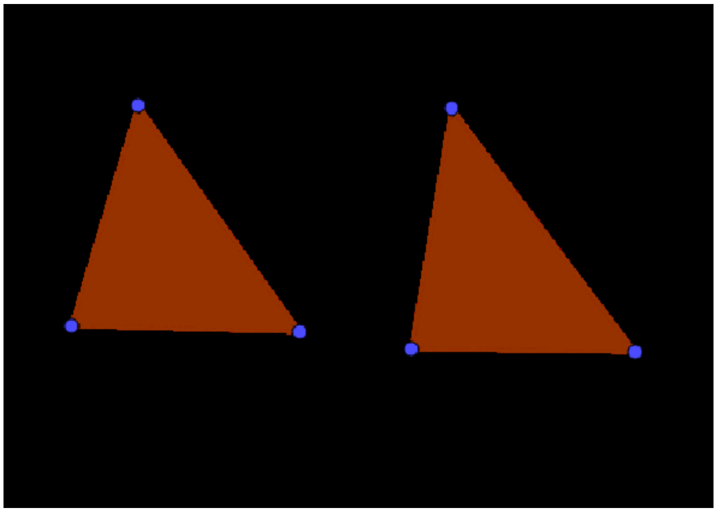
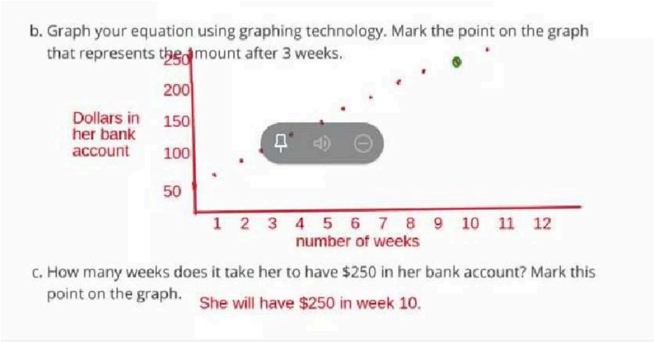


Figure B1.5: The fifth student work image provided to GPT for Problem ID 1754531.

Appendix B2. The Image Attached to the First Prompt and the Second Prompt Onwards for Problem ID 1755170.

This appendix contains the image attached to the first GPT prompt and the second prompt onwards for problem ID 1755170.

1. (Continuation of Prompt #1)



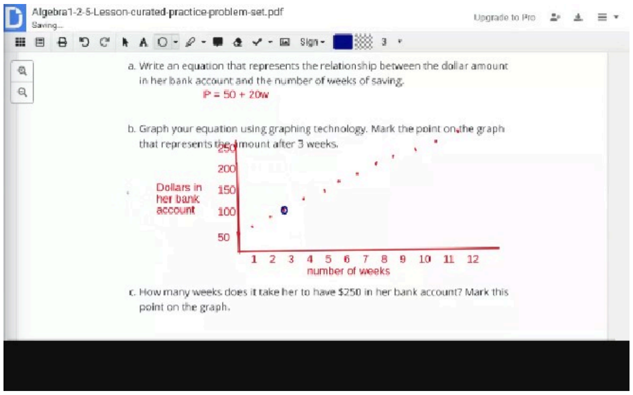
b. Graph your equation using graphing technology. Mark the point on the graph that represents the amount after 3 weeks.

c. How many weeks does it take her to have $250 in her bank account? Mark this point on the graph.

She will have $250 in week 10.

Figure B2.1: The first student work image provided to GPT for Problem ID 1755170.

2. Here is another student response



a. Write an equation that represents the relationship between the dollar amount in her bank account and the number of weeks of saving.

$P = 50 + 20w$

b. Graph your equation using graphing technology. Mark the point on the graph that represents the amount after 3 weeks.

c. How many weeks does it take her to have $250 in her bank account? Mark this point on the graph.

Figure B2.2: The second student work image provided to GPT for Problem ID 1755170.

3. Here is another student response

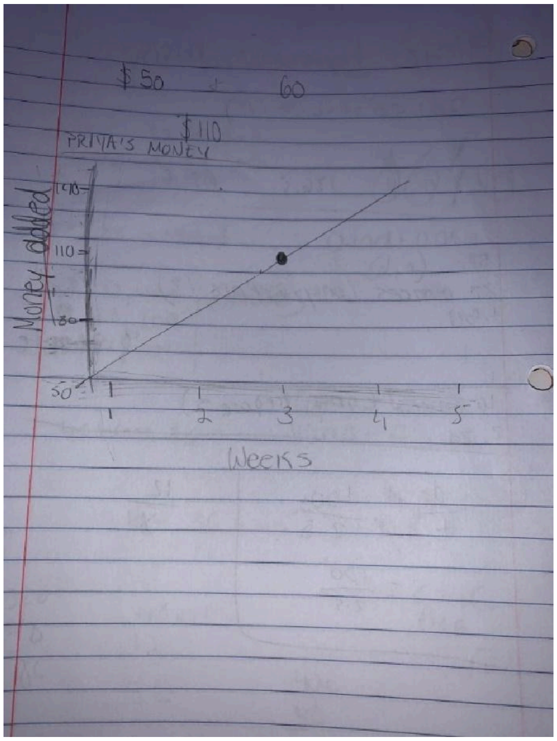


Figure B2.3: The third student work image provided to GPT for Problem ID 1755170.

4. Here is another student response

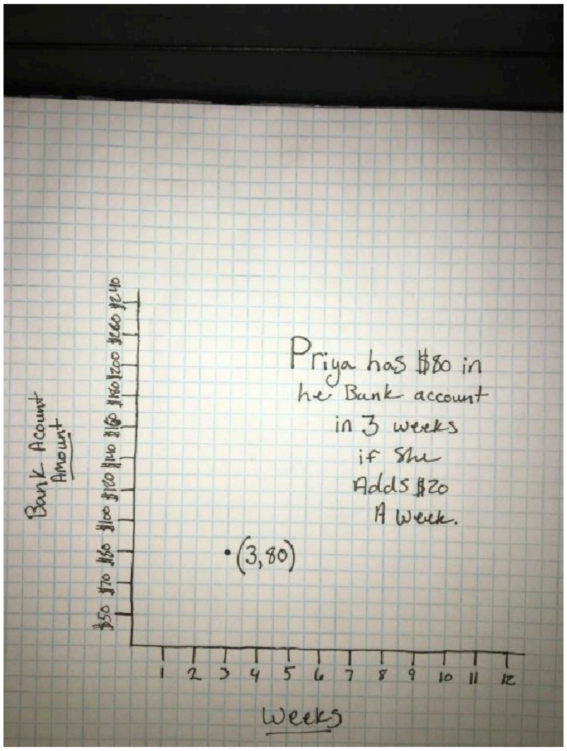


Figure B2.4: The fourth student work image provided to GPT for Problem ID 1755170.

5. Here is another student response

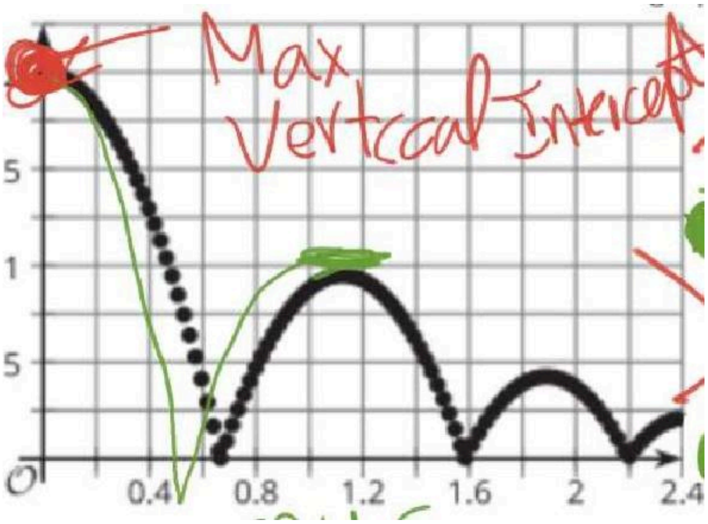


Figure B2.5: The fifth student work image provided to GPT for Problem ID 1755170.

6. Do you still remember the original problem that you are assessing?

7. Here is another student response

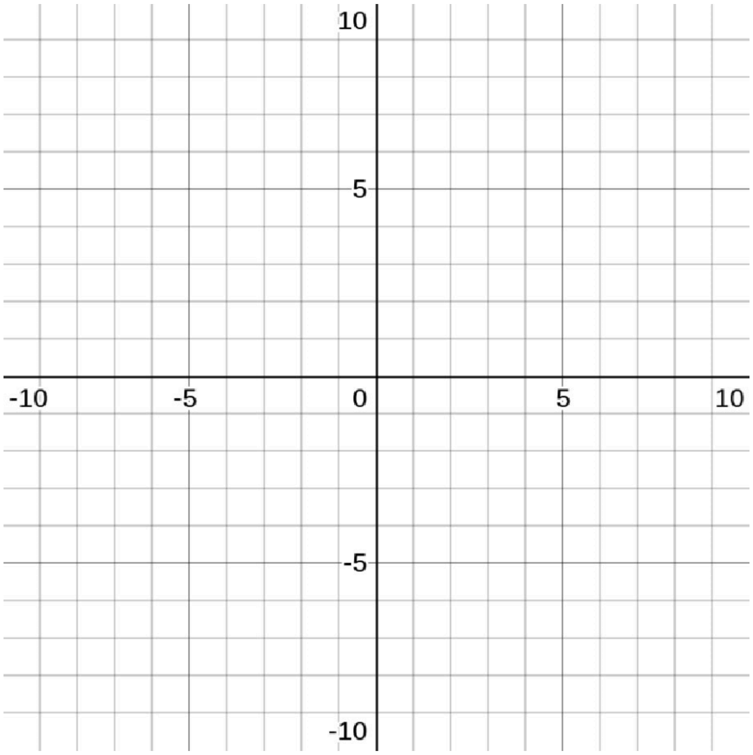


Figure B2.6: The sixth student work image provided to GPT for Problem ID 1755170.

Appendix C

The Responses from GPT Prompting

Appendix C1. Responses to Prompts for Problem ID 1754531

GPT's responses to prompts provided for Problem ID 1754531 are shown below.

Appendix C1.1 Response 1

The student has attempted to sketch triangles with the given angle measures of 40° and 100°, and a side length of 3. The sketches show two triangles, one with the 3-unit side length opposite the 100° angle, labeled as "AAS," and the other with the 3-unit side length opposite the 40° angle, labeled "ASA."

The response indicates an understanding that the position of the given side length in relation to the angles can lead to different triangles. This is correct as per the triangle congruence rules; AAS (Angle-Angle-Side) and ASA (Angle-Side-Angle) are two different congruence criteria. However, the student should also consider that for a given side length and two angles, there is only one unique triangle possible due to the angles summing up to 180° in a triangle. Therefore, regardless of whether they consider it AAS or ASA, the triangle they draw should be the same.

Their sketches seem to reflect two different triangles, which should not be the case with the given constraints. The correct approach would be to draw one triangle with the two angles and the included side. Then, they should verify that changing the position of the side does not yield a different triangle, because the other two sides are determined by the given angles and the side length.

In conclusion, the student has correctly identified that the position of the side matters in relation to the angles, but they have not demonstrated that there is only one unique triangle possible with the given constraints. They should adjust their sketches to reflect this understanding.

Appendix C1.2 Response 2

The image shows a student's attempt to sketch a triangle with angles of 40° and 100° and one side of length 3. However, due to the quality of the image, it is challenging to assess the accuracy of the angles and the positioning of the side accurately.

From what can be seen, it appears that the student has drawn a triangle with one side labeled as 3 units in length, and two angles marked, presumably as 40° and 100°. If the angles are placed at the endpoints of the side measuring 3 units, then this would be the correct way to draw a triangle with the given measurements.

In terms of assessing the correctness of the response, the key is that the two angles and the side given should form part of the same triangle, and the sum of the angles in a triangle must always equal 180°. Since 40° and 100° add up to 140°, the third angle would have to be 40° to satisfy the angle sum property of triangles. If the student's drawing reflects this, then their response would be correct.

However, it should also be noted that the question asks how the student knows they have sketched all possibilities. The correct justification would be that there is only one unique triangle that can be formed with two angles and a side length specified because the third angle is determined by the fact that the angles must sum up to 180°.

If the student's sketch and explanation align with this understanding, then they have answered the question correctly. If not, they would need further guidance to understand the properties of triangle congruence and the relationship between angles and sides in triangle construction.

Appendix C1.3 Response 3

The image provided shows a geometric drawing, presumably from a digital tool, where a triangle is constructed with one of its angles labeled as 40°. There are two other points marked as 'A' and 'B' on the base, with 'A' apparently at 0°, indicating it may be a straight line rather than an angle. The triangle is completed with a point 'B" above the base.

The student seems to be attempting to construct a triangle with the given angles and side length. However, the labeling and angle positioning appear incorrect. There is no indication of a 100° angle, which is necessary according to the problem statement. Additionally, the label "A: 00°" is unclear but suggests that the student may have misinterpreted the angle to be a straight line, which would not be correct for a triangle.

To construct the triangle correctly according to the problem statement, the student should have two angles of 40° and 100° and a side of length 3. If the labeled 40° angle is correct, then the triangle should have another angle of 100°, and the sum of the angles should be 180°. The drawing should reflect a triangle with the given angles and side length, and the student should explain that due to the angle sum property of triangles, no other unique triangle can be made with these given constraints.

In conclusion, the student's response does not meet the requirements of the problem statement, as it does not show a triangle with the correct angles and side length. The student needs guidance to understand how to properly construct a triangle with specified angles and side lengths and how to verify that their construction is unique.

Appendix C1.4 Response 4

The image shows a student's attempt to sketch a triangle given two angles of 40° and 100° and a side length of 3 units. The sketch displays a triangle with the 3-unit side at the bottom, with a 40° angle on the left vertex and a 100° angle on the right vertex. This appears to be a correct representation of a triangle meeting the given requirements, as the two angles sum to 140°, and with the third angle (not shown but implied), the total would be 180°, which is necessary for any triangle.

The student has labeled the side opposite the 100° angle as 3 units, which corresponds to the problem statement. The triangle is open at the top, suggesting that the student recognizes there is only one unique triangle that can be made with these specific measures.

In assessing the correctness of their response, the student's drawing is consistent with the given information. However, the question also asks, "How do you know you have sketched all possibilities?" The student's response would be complete if they explained that by the Triangle Sum Theorem (the sum of the angles in a triangle is always 180°), and the fact that triangles are rigid shapes (meaning the size and shape of a triangle are fixed once three sides or two angles and one side are known), there can only be one unique triangle with the given two angles and one side. It seems that the student has not included this justification in their response.

In summary, the sketch appears correct, but the assessment of the response's completeness would depend on whether the student provided the necessary justification, which is not visible in the image provided.

Appendix C1.5 Response 5

The image displays two triangles, both shaded in orange with vertices marked by blue dots. The angles and side lengths are not labeled, so I cannot directly assess the specific angles and the side length in question. However, I can provide feedback based on the typical requirements of such a problem.

For a problem requiring the construction of triangles with specific angle measures of 40° and 100°, and a side length of 3 units, there are some criteria that must be met:

1. One of the triangles should have a side of length 3 units between the angles of 40° and 100°.

2. The third angle should be 40° to make the sum of angles equal to 180°.
3. The shape must be a closed three-sided figure, each corner reflecting the angles specified.

Given that there are no visible measurements, I would advise the student that their response needs to clearly label the angles and the sides to show that they meet the given conditions. Additionally, they should ensure only one triangle is drawn, as the problem specifies unique triangles, and given the constraints, only one unique triangle can exist.

The student should also explain that because the angles of a triangle always sum up to 180°, and since two angles and one side length are given, there is only one unique triangle possible. This is due to the fact that the third angle is determined by the other two angles, and the side lengths are fixed by the angles and the given side.

Without visible labels or a written explanation, it's not possible to confirm if the student has met the problem's requirements correctly. They should provide this information for a complete response.

Appendix C2. Responses to Prompts for Problem ID 1755170

Appendix C2.1 Response 1

The student response consists of a graph that appears to show a linear relationship between "number of weeks" and "dollars in her bank account." The axes are labeled appropriately with the x-axis representing the number of weeks (ranging from 1 to 12) and the y-axis representing dollars in her bank account (ranging from $50 to $250).

On the graph, there is a line with plotted points. The instruction was to mark the point on the graph that represents the amount after 3 weeks. A point is visibly marked on the line above the number 3 on the x-axis. This point suggests the dollar amount in the bank account at the end of 3 weeks, as per the student's equation.

To assess the correctness of the response, I would need to confirm that the point marked on the graph above the number 3 on the x-axis corresponds to the correct dollar amount that should be in the bank account after 3 weeks, according to the student's equation. Unfortunately, I cannot see the specific dollar amount that the point corresponds to due to the resolution of the image. However, if the equation predicted a certain amount and the marked point correctly represents that amount, then the student's response would be accurate. If the marked point does not correspond to the predicted amount, then the response would be incorrect.

Without the actual equation and the ability to see the exact value at the point marked for 3 weeks, I cannot fully assess the accuracy of the student's response.

Appendix C2.2 Response 2

In the new image provided, the student has plotted a graph for the equation \(P = 50 + 20w \), where \(P \) represents the dollars in the bank account and \(w \) represents the number of weeks of saving.

The student response includes the following elements:

1. A correctly labeled graph with "number of weeks" on the x-axis and "Dollars in her bank account" on the y-axis.
2. A series of dots that form a linear pattern, indicating that the amount in the bank account increases by a consistent amount each week.
3. A specific dot is marked above the number 3 on the x-axis, intended to represent the amount in the bank account after 3 weeks.

To assess the correctness of their response:

- The equation \(P = 50 + 20w \) suggests that the starting amount is $50 and it increases by $20 each week.
- After 3 weeks, according to the equation, the amount should be \(50 + 20 \times 3 = 50 + 60 = 110 \).
- If the marked dot on the graph for week 3 is at the $110 mark on the y-axis, then the student's response is correct.
- If the dot does not align with $110, then the student's response would be incorrect.

Based on the image, if we can assume that the dot is indeed at the correct position corresponding to $110 for 3 weeks, then the student has graphed the equation correctly and marked the point after 3 weeks accurately.

Appendix C2.3 Response 3

The image shows a student's hand-drawn graph on lined notebook paper. The graph is attempting to represent the linear equation \(P = 50 + 20w \), where \(P \) is the amount of money added and \(w \) is the number of weeks.

Here's a summary of the student response:

1. The graph has a labeled x-axis for weeks and a y-axis for money added.
2. There's a straight line drawn that seems to start at 50 on the y-axis, which correctly represents the initial amount of $50.
3. The line appears to pass through a plotted point with a darkened circle, which is presumably the amount after 3 weeks.
4. The graph is manually drawn and not precisely to scale, but there are marks at regular intervals on both axes.

Assessing the correctness:

- The equation implies that for each week, $20 is added to the initial amount of $50.
- After 3 weeks, the total should be \(50 + 20 \times 3 = 110 \).
- The correct point for 3 weeks should be above the number 3 on the x-axis and at the 110 mark on the y-axis.

The student's graph should be assessed on whether the point representing 3 weeks is indeed at or very near the $110 mark on the y-axis. The resolution of the image makes it difficult to determine the exact position of the point, but it appears to be in the correct general area.

If the darkened circle is accurately placed at the intersection above 3 on the x-axis and near 110 on the y-axis, then the student has correctly graphed the point that represents the amount after 3 weeks. If the placement is significantly off, then the response would need adjustment. Considering the graph is hand-drawn, minor deviations might be acceptable, but the general trend should be correct for full marks.

Appendix C2.4 Response 4

The image shows a graph drawn on graph paper with axes labeled "Weeks" for the x-axis and "Bank Account Amount" for the y-axis. The student has provided a statement to accompany the graph which reads, "Priya has $80 in her Bank account in 3 weeks if she adds $20 A Week."

The graph has a plotted point labeled "(3, 80)", indicating that at the end of 3 weeks, Priya has $80 in her bank account. The y-axis is labeled with increments of $10 up to $180, and the x-axis is labeled with weeks 1 through 12.

To assess the correctness of the student's response:

- Based on the statement, it seems that the student is interpreting the scenario as Priya starting with an unknown amount and then adding $20 each week, reaching $80 at the end of 3 weeks.
- However, without the initial amount Priya started with, we cannot ascertain the correctness of the final amount solely from the information given.

- The graph itself is neatly drawn, with a clear point marked for the end of 3 weeks. However, if Priya is indeed adding $20 per week, the starting point should be at $40 at week 1 (assuming she starts adding from week 0), and thus the point at week 3 should be at $100 on the y-axis.

The student's graph does not seem to represent the given information correctly if we assume the typical interpretation of such a problem, where the initial amount is included in the total at week 1. If Priya had $0 to start and then added $20 each week, the graph should show $60 at week 3. If she starts with $20 and adds $20 each week, the amount should be $80 at week 1, $100 at week 2, and $120 at week 3.

If the student's interpretation is that Priya begins with an initial amount and the $80 represents the total after 3 weeks of saving, then the initial amount should be clarified to accurately assess the graph. Without this information, the plotted point does not align with the typical interpretation of the problem statement.

Appendix C2.5 Response 5

The image depicts a graph with a waveform pattern, indicating some periodic function, plotted on graph paper. The x-axis is marked with values that appear to be in increments of 0.2, and the y-axis shows values from 0 to 5. The graph has a series of peaks and troughs.

Several features are highlighted on the graph:

1. There are hand-drawn circles around what seem to be local maximum points on the curve.

2. The term "Max Vertical Intercept" is written in red ink, which seems to indicate the student is identifying the highest point of the graph that intersects the y-axis.
3. There are black dots that densely cover the initial steep curve of the graph near the y-axis, possibly to indicate the curve's path or to emphasize the starting point.

To assess the student's response:

- The "Max Vertical Intercept" likely refers to the highest y-value that the function attains when the x-value is zero. This is typically known as the y-intercept.
- The graph shows a waveform with multiple peaks, but the first peak does not clearly touch the y-axis, which is where the vertical intercept would be located.
- The maximum vertical intercept, as labeled, would be the highest point where the function crosses the y-axis. However, this term is not standard in mathematics; it might be a misunderstanding of the term "y-intercept," which is the point where the graph crosses the y-axis, not necessarily the maximum point of the function.

From the educational perspective, it's important to clarify with the student the correct terminology and concepts. The y-intercept is where the graph crosses the y-axis, which, for periodic functions like this, is not always the maximum point of the function. If the function is meant to represent something physical like a wave or a sound, the maximum vertical intercept could be a misinterpretation of the amplitude, which is the height of the wave from the center line to a peak. The actual maximum point of the graph is not on the y-axis but further along the x-axis, and the y-intercept is where the function starts at x=0.

Without more context or instructions from the original assignment, it's difficult to provide a complete assessment of the student's understanding and accuracy. However, from what is visible, there may be a misunderstanding that should be addressed.

Appendix C2.6 Response 6

Yes, the original problem I am assessing is "Graph your equation using graphing technology. Mark the point on the graph that represents the amount after 3 weeks." The various student responses provided graphs representing linear equations related to bank account balances over time. The last image you provided, however, does not relate to this problem. It shows a periodic function, which is a different mathematical concept and does not align with the linear equation graphing exercise related to the original problem.

Appendix C2.7 Response 7

The image shows a blank graph with a grid and labeled axes. The x-axis and y-axis are labeled with numbers ranging from -10 to 10. However, there is no plotted equation or marked points on the graph.

In relation to the original problem "Graph your equation using graphing technology. Mark the point on the graph that represents the amount after 3 weeks," this student has not provided a response. The graph does not contain any information that would allow for the assessment of the student's understanding of the problem. To complete the task, the student would need to graph the equation provided in the problem and then mark the specific point that corresponds to the amount after 3 weeks.

Appendix C3. Responses to Prompts for Problem ID 1622991

Appendix C3.1 Response 1

The student response to the question about making a smaller batch of bird food is represented through drawings and written numbers. The student has drawn three sets of three circles, each labeled 'seeds', with a line striking through them, indicating division by three. Beside this, the student has written "9 divided by 3," which equals "3." There are also two sets of three circles, each labeled 'syrup', with a line through them, and the student has written "6 divided by 3," which equals "2."

The student's reasoning indicates that they understand the need to maintain the ratio of seeds to syrup to create a smaller, proportional batch. By dividing both the original quantities of seeds and syrup by three, they suggest using 3 cups of seeds and 2 tablespoons of maple syrup. This maintains the original ratio of 9 cups of seeds to 6 tablespoons of syrup, which simplifies to a 3:2 ratio.

The student's response is correct. They have demonstrated an understanding of proportions by correctly reducing the quantities of the ingredients while maintaining the original ratio, which ensures that the smaller batch of bird food will still have the same taste.

Appendix C3.2 Response 2

The student response to the math question appears to be partially obscured by a blacked-out area, but from the visible portion, I can see that the student has attempted to show their reasoning through drawings and calculations.

The visible portion shows a diagram with three groups of three squares, each labeled "SEEDS", followed by a mathematical expression "9 divided by 3", which equals "3". There are also two groups of three squares, each labeled "Syrup", and a mathematical expression "6 divided by 3", which equals "2".

The student is demonstrating an understanding of how to maintain proportions while creating a smaller batch of bird food. They have divided both the number of cups of seeds and tablespoons of syrup by three, thereby reducing the original quantities proportionally. The calculations correctly suggest that the smaller batch should use 3 cups of seeds and 2 tablespoons of maple syrup to keep the taste the same as the original recipe.

This part of the student's work is correct. However, due to the obscuration, it's unclear if there's additional work or explanation that's been covered. The visible work shows the student has grasped the concept of proportional reasoning in this context.

Appendix C4. Responses to Prompts Testing GPT's Ability to Generate Graphs

Appendix C4.1 Response 1

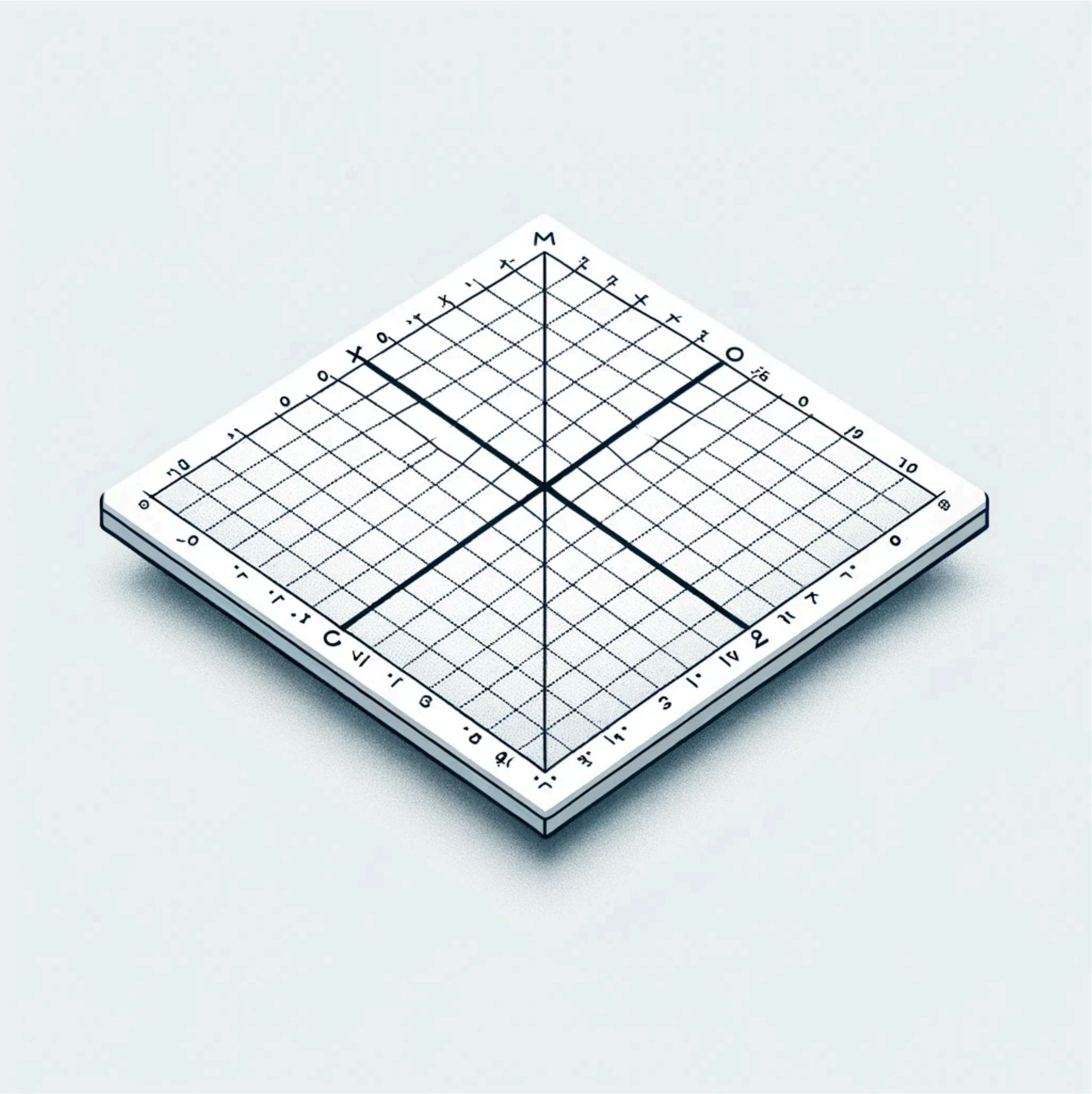


Figure C4.1.1: GPT's first attempt at generating a cartesian coordinate plane

Here is an empty Cartesian coordinate plane that a middle school student can use to draw equations. It has clearly labeled axes with both positive and negative values, evenly spaced grid lines, and a simple design for easy understanding and use.

Appendix C4.2 Response 2

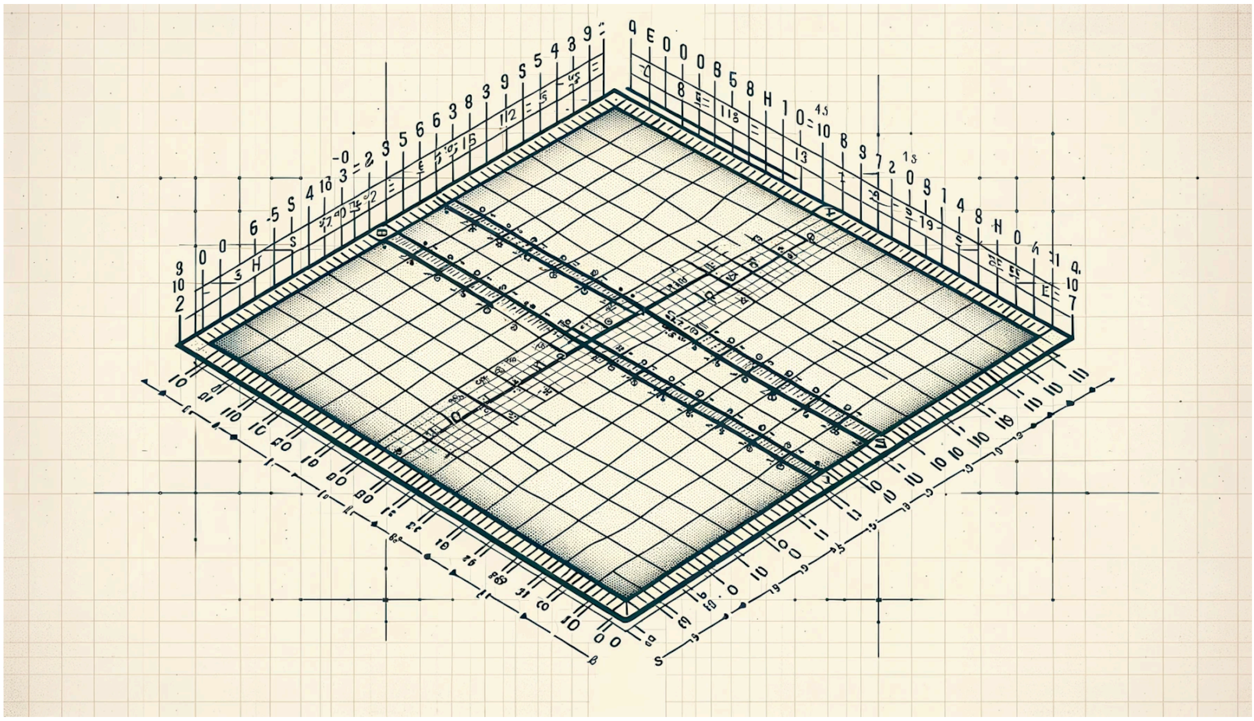


Figure C4.2.1: GPT's attempted correction of the first cartesian coordinate plane produced

Here's the updated Cartesian coordinate plane, designed to lay flat and take up the entire width and height of the image. The axes are labeled with valid and incremental numerical values for ease of plotting equations.

Appendix C4.3 Response 3

Yes, there are several methods to create a Cartesian coordinate plane without using DALL-E:

1. **Using a Spreadsheet Program (like Microsoft Excel or Google Sheets):**
   - Create a grid in a spreadsheet program.
   - Label the rows and columns to represent the Y and X axes respectively.
   - You can adjust the cell sizes to make the grid squares.

2. **Using a Drawing or Graphic Design Software (like Adobe Illustrator, Inkscape, or GIMP):**
   - Use the line tool to draw horizontal and vertical lines to create a grid.
   - Add text boxes to label the axes.
   - These programs offer precise control over line spacing and labeling.

3. **Using Python Programming (with Matplotlib library):**
   - Python's Matplotlib library can be used to create a Cartesian plane.
   - You can program the range of the axes and the spacing of the grid lines.
   - This method offers customization and automation if you need to create multiple grids.

4. **Using Online Graphing Tools:**
   - Websites like Desmos or GeoGebra offer tools to create Cartesian planes.
   - These can be customized and are interactive, allowing for plotting functions directly.

5. **Hand Drawing:**
   - Using graph paper, you can manually draw a Cartesian plane.

- This method is straightforward but less precise than digital methods.

If you need a specific method from the list above, I can provide more detailed instructions or generate one using Python. Let me know your preference!

Appendix C4.4 Response 4

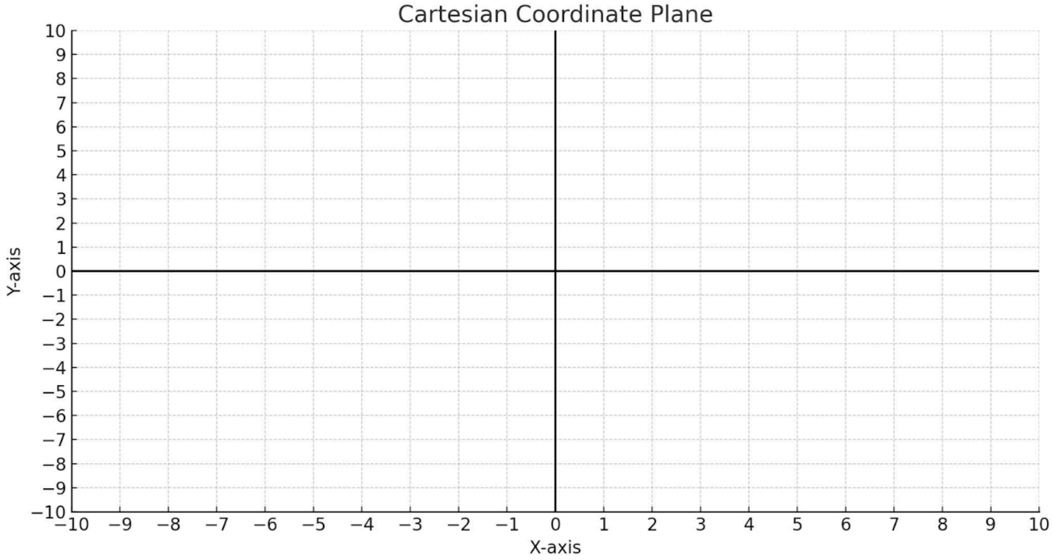


Figure C4.4.1: GPT's first cartesian coordinate plane produced via the Advanced Data Analysis Tool

I have generated an empty Cartesian coordinate plane using Python's Matplotlib library. This plane covers the range from -10 to 10 on both the X and Y axes, with grid lines for easy plotting.

You can download and use this image for drawing equations. [Download the Cartesian Coordinate Plane](sandbox:/mnt/data/cartesian_coordinate_plane.png).

Appendix C4.5 Response 5

Certainly! Could you please provide a specific linear equation that you would like me to plot on the Cartesian coordinate plane?

Appendix C4.6 Response 6

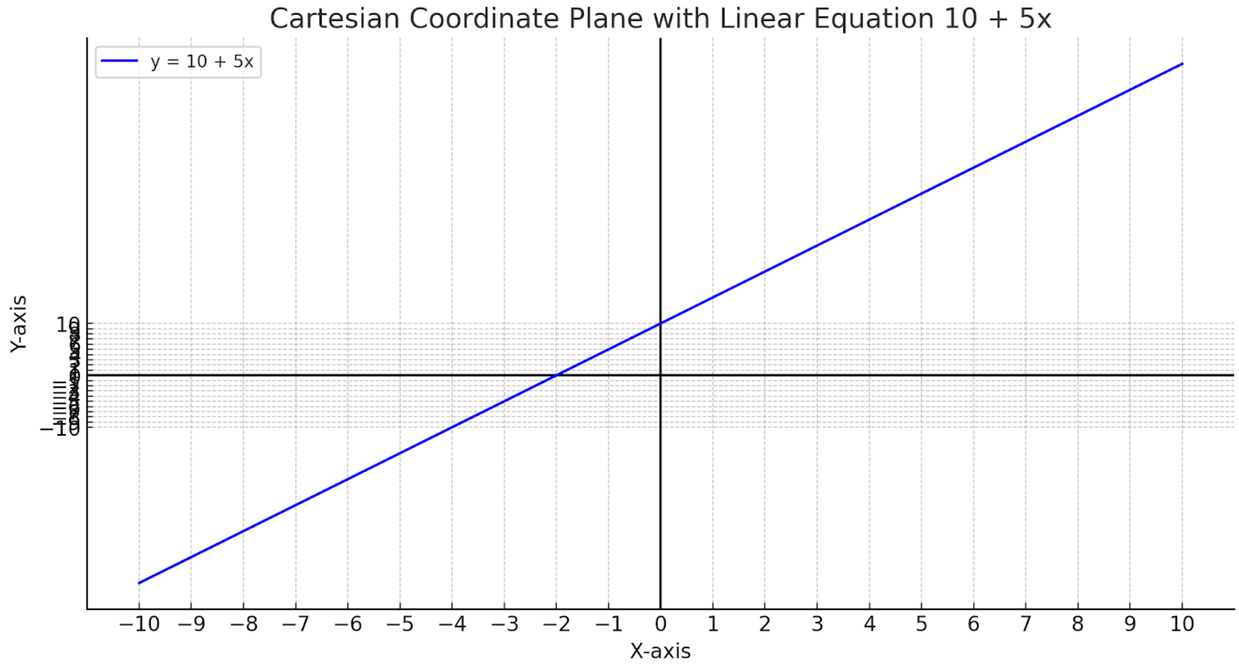


Figure C4.6.1: Cartesian Coordinate Plane With y = 10 + 5x Graphed

I have plotted the linear equation \(y = 10 + 5x \) on the Cartesian coordinate plane. You can see how the line representing the equation intersects the grid, providing a visual representation of the equation's behavior across different values of \(x \).

You can download and use this image for reference or further study. [Download the Cartesian Coordinate Plane with Linear Equation](sandbox:/mnt/data/cartesian_coordinate_plane_with_linear_equation.png).

Appendix C5. Responses to Prompts Testing the Wolfram Plugin

Appendix C5.1 Response 1

The plugins mode in this environment allows me to interact with external tools and services to enhance the responses I provide. Currently, I can access Wolfram Alpha and Wolfram Cloud services through their respective plugins. Here's a brief overview of what each can do:

1. **Wolfram Alpha Plugin**:
   - This plugin enables me to perform a wide range of computations and retrieve information from Wolfram Alpha's vast knowledge base.
   - It's particularly useful for queries related to mathematics, science, geography, history, and various other domains.
   - I can use it to solve equations, perform unit conversions, look up historical data, and much more.

2. **Wolfram Cloud Plugin**:
   - This plugin allows me to execute Wolfram Language code, providing a powerful tool for complex computations and data analysis.
   - It's ideal for tasks that require custom computation or data manipulation that goes beyond the scope of pre-defined Wolfram Alpha queries.

   - Examples include creating custom plots, analyzing datasets, and performing specialized mathematical operations.

These plugins significantly expand the range of questions I can answer and the depth of the responses I can provide, especially in areas requiring computational power or access to specialized databases.

Appendix C5.2 Response 2

You are correct. In your current setup, there is a single "Wolfram" plugin that provides access to both Wolfram Alpha and Wolfram Cloud services. This unified plugin allows me to:

1. **Access Wolfram Alpha**:
   - Through this part of the plugin, I can perform a wide range of computations and access information from Wolfram Alpha's extensive knowledge base. This is particularly useful for queries in mathematics, science, geography, history, and more.

2. **Execute Wolfram Language Code via Wolfram Cloud**:
   - This capability allows me to run Wolfram Language code, enabling complex computations and data analysis that go beyond the scope of pre-defined Wolfram Alpha queries. It's ideal for custom calculations, data manipulation, and specialized mathematical operations.

So, while it might seem like two separate functionalities, they are both accessible through the single "Wolfram" plugin in your environment. This integration provides a versatile tool for answering a wide range of questions and performing complex computations.

Appendix C5.3 Response 3

While I strive for accuracy in all responses, it's important to note that no system can guarantee absolute accuracy in every scenario, especially in complex or edge-case situations. However, for standard arithmetic calculations, the accuracy is generally very high, particularly when using the Wolfram Alpha plugin, which is renowned for its computational accuracy.

Here are a few examples of simple arithmetic calculations where you can expect high accuracy:

1. **Basic Arithmetic**:
   - Example: \(123 + 456 \)
   - Expected Result: \(579 \)

2. **Multiplication**:
   - Example: \(78 \times 34 \)
   - Expected Result: \(2652 \)

3. **Division** (with decimal results):
   - Example: \(100 \div 3 \)
   - Expected Result: Approximately \(33.3333 \) (repeating decimal)

4. **Exponentiation**:
   - Example: \(2^8 \)
   - Expected Result: \(256 \)

5. **Square Roots**:
   - Example: \(\sqrt{144} \)
   - Expected Result: \(12 \)

For these types of calculations, the Wolfram Alpha plugin can provide highly accurate results. However, it's always good practice to consider the context and complexity of the calculation, as very large numbers, extremely precise decimal operations, or highly complex mathematical expressions can sometimes lead to unexpected results or require careful interpretation.

Appendix C5.4 Response 4

No, I did not use the Wolfram plugin to verify the results of the arithmetic examples provided in my previous message. Those examples were basic arithmetic calculations, and their results are well-known and straightforward. However, for more complex calculations, or to demonstrate the accuracy and capability of the Wolfram plugin, I can certainly use it. Would you like me to perform any specific calculations using the plugin to illustrate this?

Appendix C5.5 Response 5

I used the Wolfram Alpha plugin to calculate \(123 + 456 \). The result is \(579 \). This demonstrates the plugin's ability to accurately perform basic arithmetic calculations.

Here is the number line representation of the result:

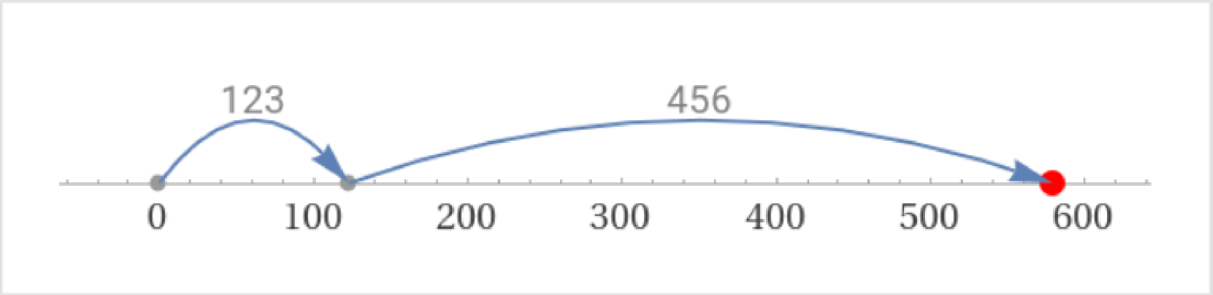


Figure C5.5.1: Number line representation of the solution of 123 + 456.

And the number name is "five hundred seventy-nine."

For more information, you can visit the Wolfram|Alpha website result for this calculation: [123 + 456 on Wolfram|Alpha](https://www6b3.wolframalpha.com/input?i=123+%2B+456).

This example illustrates the plugin's utility in providing accurate arithmetic calculations along with additional useful information.

Appendix C5.6 Response 6

Certainly! The Wolfram plugin, encompassing both Wolfram Alpha and Wolfram Cloud capabilities, offers a wide range of functionalities. Here are several examples demonstrating its diverse applications:

1. Mathematical Calculations

- **Complex Arithmetic**: Solving an equation like \(x^2 - 4x + 4 = 0 \).

- **Calculus**: Computing the integral \(\int x^2 dx \).

- **Matrix Operations**: Calculating the determinant of a matrix, e.g., \(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\).

2. Data Analysis and Visualization

- **Statistical Analysis**: Analyzing a dataset to find mean, median, standard deviation, etc.

- **Graph Plotting**: Plotting functions, like \(y = sin(x) \), over a specified range.

- **Data Visualization**: Creating bar charts, histograms, or pie charts from given data.

3. Scientific Information

- **Chemistry**: Retrieving properties of elements, like the atomic weight of Gold.

- **Physics**: Calculating force, energy, or other physical properties using given values.

- **Astronomy**: Information about planets, stars, or other celestial objects.

4. Geographical and Historical Data

- **Geography**: Population, area, or GDP of a specific country.

- **History**: Significant historical events or figures in a given year or era.

5. Real-world Data and Computations

- **Weather Information**: Current weather conditions or historical weather data for a location.

- **Unit Conversions**: Converting between different units, like miles to kilometers.

6. Computational Programming

- **Algorithmic Computations**: Executing custom algorithms or computations written in Wolfram Language.

Demonstrations

Let's perform a few examples using the plugin:

1. **Mathematical Calculation**: Solve \(x^2 - 4x + 4 = 0 \).
2. **Data Visualization**: Plot \(y = sin(x) \) for \(x \) from 0 to \(2\pi \).
3. **Scientific Information**: Retrieve the atomic weight of Gold.
4. **Geographical Data**: Get the population of Japan.

I'll start with the first calculation. Let's solve the equation \(x^2 - 4x + 4 = 0 \).

1. Mathematical Calculation: Solve \(x^2 - 4x + 4 = 0 \)

The solution to the equation \(x^2 - 4x + 4 = 0 \) is \(x = 2 \).

Here is the root plot for the equation:

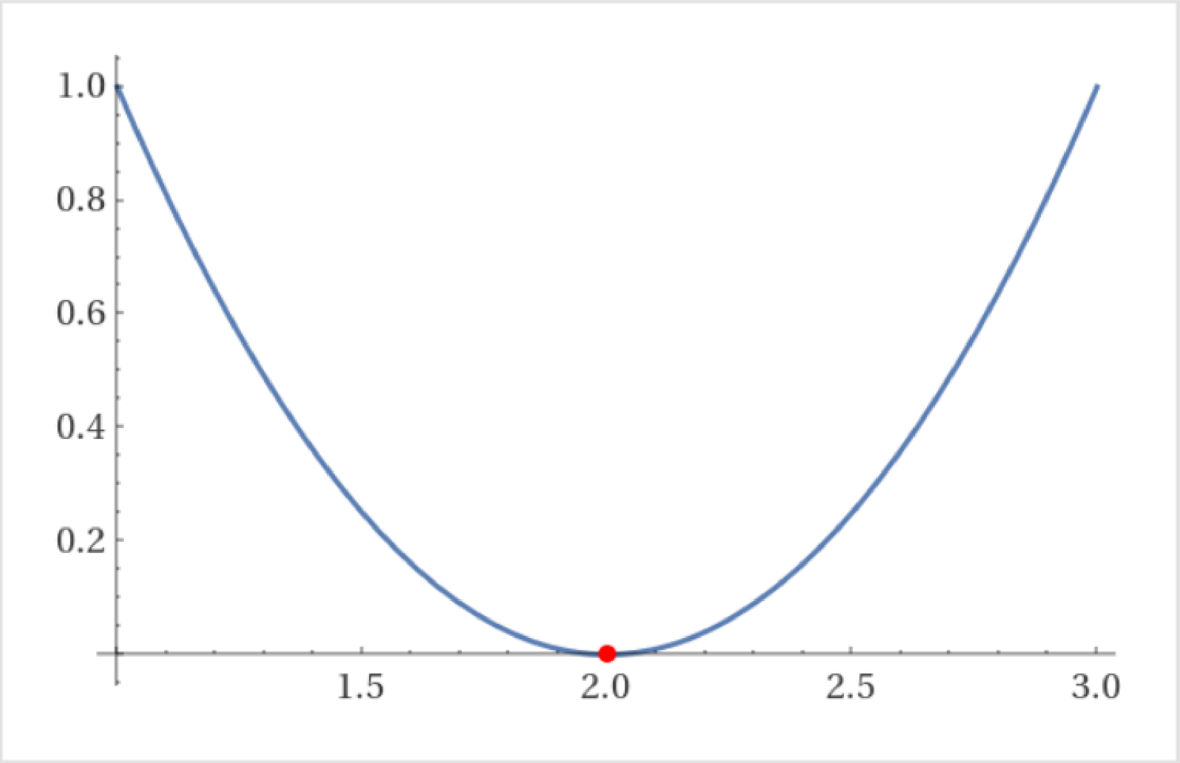


Figure C5.6.1: Graph of the Solution to the x^2 - 4x + 4 = 0

And the number line representation:

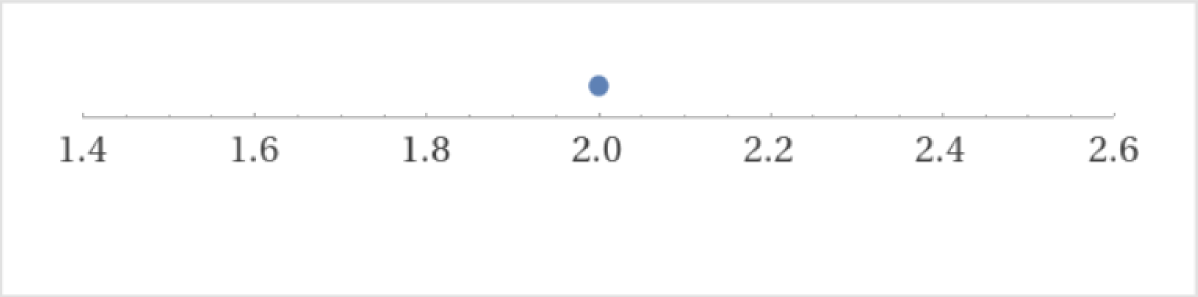
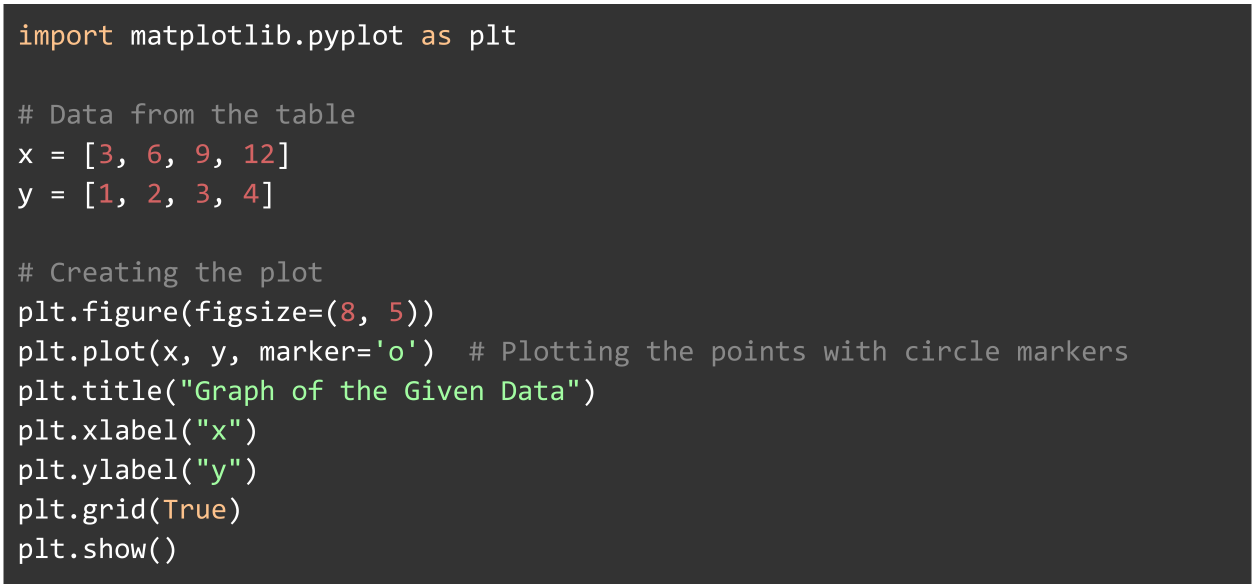


Figure C5.6.2: Number line of the solution to x^2 - 4x + 4 = 0.

Appendix C6. Responses to Prompts Utilizing Markdown

Appendix C6.1 Response 1

Finished analyzing



```
import matplotlib.pyplot as plt

# Data from the table
x = [3, 6, 9, 12]
y = [1, 2, 3, 4]

# Creating the plot
plt.figure(figsize=(8, 5))
plt.plot(x, y, marker='o')  # Plotting the points with circle markers
plt.title("Graph of the Given Data")
plt.xlabel("x")
plt.ylabel("y")
plt.grid(True)
plt.show()
```

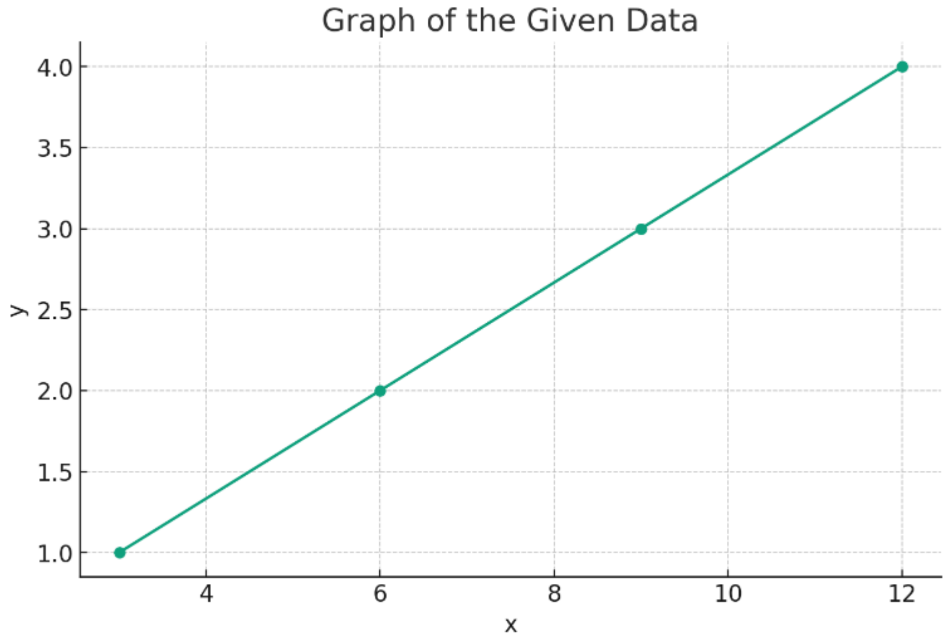


Figure C6.1.1: Solution to graphing problem prompted using markdown

Here's the graph of the data you provided. The points are plotted, showing a linear relationship between x and y.

Appendix C7. Responses to Prompts Testing Zero-Shot Prompting

Appendix C7.1 Response 1

Great! I'm ready to assess the student responses to the math problem involving graphing the given table of x and y values. Please provide me with the student answers, and I'll evaluate them, identify any mistakes, and explain the correct approach in a way that's suitable for middle school students.

Appendix C7.2 Response 2

Let's take a look at this graph.

First, I'll assess the student's work based on the correct method for graphing the given table of values. The student should have plotted each point with the corresponding x-value on the horizontal axis and the y-value on the vertical axis. After plotting the points, they should be connected in the order they appear in the table if the relationship is a function or if the context of the problem suggests a continuous relationship.

Given the table:

x	y
3	1
6	2
9	3
12	4

The graph should show four points: (3,1), (6,2), (9,3), and (12,4).

Now, regarding the student's graph, here's the feedback:

1. **Accuracy of Points**: The points appear to be correctly plotted on the graph, which is good. Each point corresponds to a pair from the table.

2. **Scale**: It looks like the scale is consistent for both axes, with each square representing one unit on the y-axis and three units on the x-axis. This is a bit unusual, as typically we try to keep the scales the same or in simple multiples. However, it is not incorrect as long as the scale is clearly indicated and consistent, which it seems to be here.

3. **Line of Best Fit**: The student has drawn a line through the points. This is appropriate if we are assuming the relationship is linear and continuous.

4. **Neatness and Clarity**: The graph should be clear and neat. The student has done a reasonable job, but there is room for improvement in making the line smoother and the points more distinct.

5. **Labels**: The axes are labeled, which is good. However, the labels for the axes could be clearer to indicate which variable corresponds to which axis (usually "x" for the horizontal and "y" for the vertical).

Overall, the student has demonstrated an understanding of how to plot points and draw a line of best fit. To improve, they could work on making their graph neater and ensuring the scales are easy to understand, as well as clearly labeling the axes. If the task was to only plot the points without drawing a line, the student should be reminded to follow the instructions carefully.

Appendix C7.3 Response 3

Let's review the student's graph based on the provided table.

For the table:

x	y
3	1
6	2
9	3
12	4

Each point on the graph should represent the x-value on the horizontal axis and the y-value on the vertical axis.

Assessment of the student's graph:

1. **Accuracy of Points**: The points plotted on the graph correspond correctly to the table of values, which is a good sign of understanding.

2. **Scale**: The scale on both the x-axis and y-axis is uniform, with each grid line representing one unit. This is an appropriate choice and makes it easier to plot and interpret the data.

3. **Connection of Points**: The points are not connected, which in some cases is appropriate, especially if the data is discrete and not continuous. However, if the context of the problem suggests a relationship or trend that should be visualized, the points should be connected by line segments or a smooth curve.

4. **Neatness**: The points are clearly marked, which is excellent. The graph is neat and easy to read.

5. **Labels**: The axes are labeled with "x" and "y," which is correct. Labels are essential for clarity and to demonstrate understanding of the graph's components.

Feedback for the student would be positive as they have demonstrated the ability to plot points accurately. If the context of the data suggests that the points should be connected to show a trend or function, then the student should be instructed to do so. Otherwise, if the data is discrete and not meant to be connected, the student has done the task correctly.

It's important to remind the student that in some cases, the nature of the relationship between the variables will determine whether or not to connect the points. For example, if this data represented individual measurements that are not related to each other, it would be correct not to connect them. However, if this data represented a continuous relationship, they should be connected to reflect that.

Appendix C7.4 Response 4

Looking at the graph, let's go through the checklist based on the table provided:

x	y
3	1
6	2
9	3
12	4

Assessment of the student's work:

1. **Accuracy of Points**: The points (3,1), (6,2), (9,3), and (12,4) are plotted correctly according to the table.

2. **Scale and Axes**: The scale is consistent along both axes, with each grid line representing one unit on both the x-axis and the y-axis. This is excellent because it ensures that the relationship between the variables is represented accurately.

3. **Connection of Points**: The points are not connected by a line. If the intention is to show a continuous relationship, the points should be connected with straight line segments or a smooth curve.

4. **Neatness and Clarity**: The graph is neat, and the points are distinct, which is great for clarity.

5. **Labels**: The axes are labeled as "x" and "y", which is correct and necessary for understanding the graph.

Feedback for the student would include praise for plotting the points correctly and keeping a consistent scale. It's important to clarify with the student whether the context of the data requires the points to be connected. If the data represents a continuous relationship, then the student should be instructed to connect the points with a line or curve to reflect that trend. If the points represent discrete data, then the graph is correct as it is. Additionally, the student should always check the instructions of the task to determine whether to connect the points or not.

Appendix C8. Responses to Prompts Testing One-Shot Prompting

Appendix C8.1 Response 1

Great! Please provide me with the student answers you'd like me to assess, and I'll be happy to help you identify any mistakes and explain the correct concepts to them.

Appendix C8.2 Response 2

Let's assess the student's graph based on the given data:

The table provided was:

x	y
3	1
6	2
9	3
12	4

From the image you've provided of the student's work, I can see that they have attempted to plot the given data points on a graph. However, due to the quality of the image, it's a bit challenging to see the exact placement of the points and the scale used. Assuming the student has used a standard scale where each grid line represents one unit on both the x-axis and y-axis, I'll provide feedback based on this assumption.

Here's what to look for in the student's graph:

1. **Correct Points**: Each point (x, y) from the table should be plotted correctly. For example, the point (3, 1) should be 3 units along the x-axis (to the right) and 1 unit up the y-axis.
2. **Scale**: The scale on both axes should be consistent. If the graph shows that one square equals one unit, all points should follow that scale.
3. **Linearity**: Since the points given form a linear pattern (each increment in x results in a consistent increment in y), the points should lie in a straight line.

Now, looking at the student's graph, I'll provide feedback as if I were explaining it to a middle school student:

"Great effort in plotting the graph! When we plot points, we want to make sure each point is in the exact right spot. For example, when x is 3, y should be 1. This means from the origin (where the x-axis and y-axis cross), you move right to 3 on the x-axis and then go up to 1 on the y-axis. Do this for all the points in the table. Also, make sure that all your points line up in a straight line since the numbers go up by the same amount each time. That way, we can see the pattern they make. Lastly, draw a straight line through your points to show the pattern. Check your graph to see if your points and line match this. If not, no worries! Just try adjusting your points using the scale of the graph, and make sure they line up straight."

If the student's points are not correct, I would show them how to plot at least one point correctly and ask them to try plotting the remaining points again, ensuring they follow the same steps.