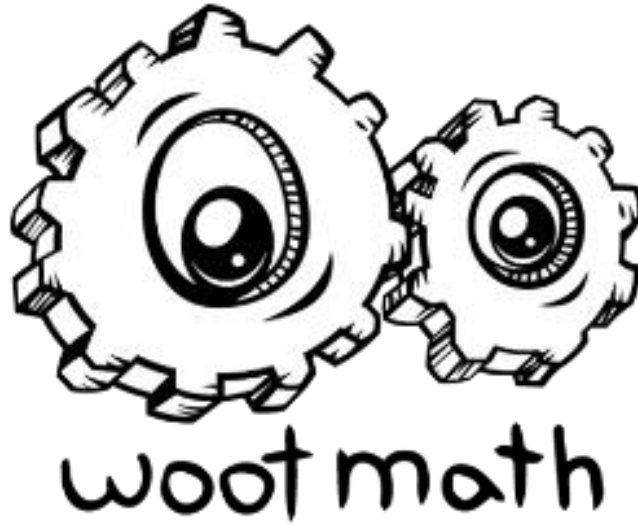


Woot Math 2



Content and digitization

By: Dante Arcuri, Thom Deisz, Toby Crocker, Cory Kennedy

Date: June 19th, 2019

Introduction:

Woot Math is a six-year-old startup located in Boulder, CO. Their mission is to “bring mathematics to life, closing the gap” by offering engaging, research-backed solutions to math classrooms everywhere using custom formative assessment, actionable data, and personalized learning. With the intention of leveraging technology to empower and assist math teachers. The Client, Woot Math, represented by Brent Milne and Sean Kelly, has expressed their need for a single-page web application, that is a web app that rewrites the current page as opposed to loading a separate page from a web server, capable of uploading math worksheets (stored as an image or a pdf) and segmenting them into sections that make intuitive sense (i.e. by question/answer, ASCII text, handwritten text, graphs/charts, equations, etc.), including an interface that enables human input where necessary or desired, and a backend database connection. They have specified that the user interface be implemented using Vue.js, supported by a TypeScript RESTful API, linked to a MongoDB database and the Optical Character Recognition (OCR) / Computer Vision (CV) tools necessary to support the above requirements, via a Node.js/Express server architecture. The expectation for our team was to conduct research into the initial design of this application, our primary objective being research into available libraries and computer vision tools that can be used to implement the desired image segmentation objectives. A dockerized environment will be built out by the client as our project progresses to teach and familiarize our web team with the process of developing a web application meant to scale, and creating the tools necessary for allowing devs to collaborate on the project regardless of their operating system and package collection. Based upon our initial meetings with the client and their list of desired features for the finished product, our strategy is to lay groundwork for

them to build upon, conducting the research that will support the implementation of as many of the identified requirements for the finished product as possible during the length of our contract.

Requirements and Features:

Following discussions with our client and through our assessment of the required deliverables, our team adopted an exploratory and research based approach as opposed to the Agile/iterative strategy discussed in class that adheres to a rigid set of requirements. That said, we've compiled a list of functional requirements that are general enough to allow the flexibility that our client desires, yet specific enough to be testable.

Collect Document Submissions:

Support user ability to upload a pdf, jpeg, or png. This may include images taken by the user, requiring keystone correction (rectifying image distortion) and cleaning. However, the computer vision techniques used in segmenting and optical character recognition may not be accurate, due to the challenge of noise reduction prior to segmentation. Though we initially implemented an algorithm to provide the keystone correction, the difficulty in reducing noise prior to segmentation obliged us to deprioritize this deliverable and focus on "clean" documents.

Human-Assisted Segmentation:

Using computer vision libraries and known algorithms, such as x-y cut, the program will return an initial set of suggested cut locations to the user, represented as lines over-layed on the image. The user may then adjust the position of the proposed cut-lines to ensure the document will be segmented in the desired fashion. This data will be sent to the image processing backend via the restful api for final document segmentation. The resulting segments will then be returned to the user who can choose to delete or keep particular segments. Those segments can then be tagged and edited. Our team was able to implement an algorithm to handle segmentation at the problem level, and a UI that supports dynamically placing and tracking the cutlines, however we were unable to complete the plumbing (the connection between front and back end) necessary to bind these features together; Plumbing being the interconnectivity of the different components of the program. Our client preferred to take the lead on routing between the restful api and the image processing scripts. Unfortunately, updates to the dockerized environment were delayed beyond our capacity to complete the plumbing for this feature on the back end.

Identifying Unique Sub-Segments:

Math worksheets are typically composed of ASCII characters, graphs, charts, and equations, as well as handwritten text. We anticipate the usefulness of breaking these unique

entities apart and analyzing them separately, therefore our image processing component should be able to discern these subsections below the problem-level segmentation. This feature might be particularly useful for extracting question/answer pairs. Our team conducted extensive research into the Google.cloud CV API, SciKit Learn, and OpenCV. Of the three Google.cloud proved the most promising, and we were able to implement an effective subsegmenting algorithm. We also began cursory research into Tesseract at the request of our client, however we were unable to draw anything meaningful from it in the time we had remaining.

Digitize Content for Editing:

Following the document submission, user adjustment and approval of outlines, and the problem and sub-level segmentations, the web-app should be able to display the each problem in an HTML textarea, allowing the user to edit and modify the question. Below the textarea will be check-boxes that allow the user to add tags describing the mathematical content being displayed. The tagging feature enables teachers to search the database for problems that contain similar tags, such as geometry, algebra, fractions, or complex numbers. The edit feature might be useful for constructing test, quiz, or homework questions that are similar to problems seen on the worksheets. The implemented UI displays the appropriate HTML containers, with the capability of generating them dynamically and filling the text areas from an array of ‘problems’ fetched from the database.

Include Capability to Accept/Reject Low Certainty Segmentations:

As with all computer vision and OCR techniques we anticipate that our algorithms will not always be able to segment the document with complete accuracy. To that effect, the user must have the ability to accept properly processed portions of the image while rejecting those that introduce inaccuracies that are beyond the capability of the ‘edit segment’ feature to remedy. In the same UI component that holds the textareas and tagging capabilities described above, we would like to display the image and the confidence interval of the segmentation next to the textarea. Due to our routing troubles, this portion has not been implemented, though the UI does contain the appropriate button to bind an accept/reject value to each ‘problem’ entity in the form.

System Architecture:

The following description of our system architecture refers to the design displayed in Figure 1.

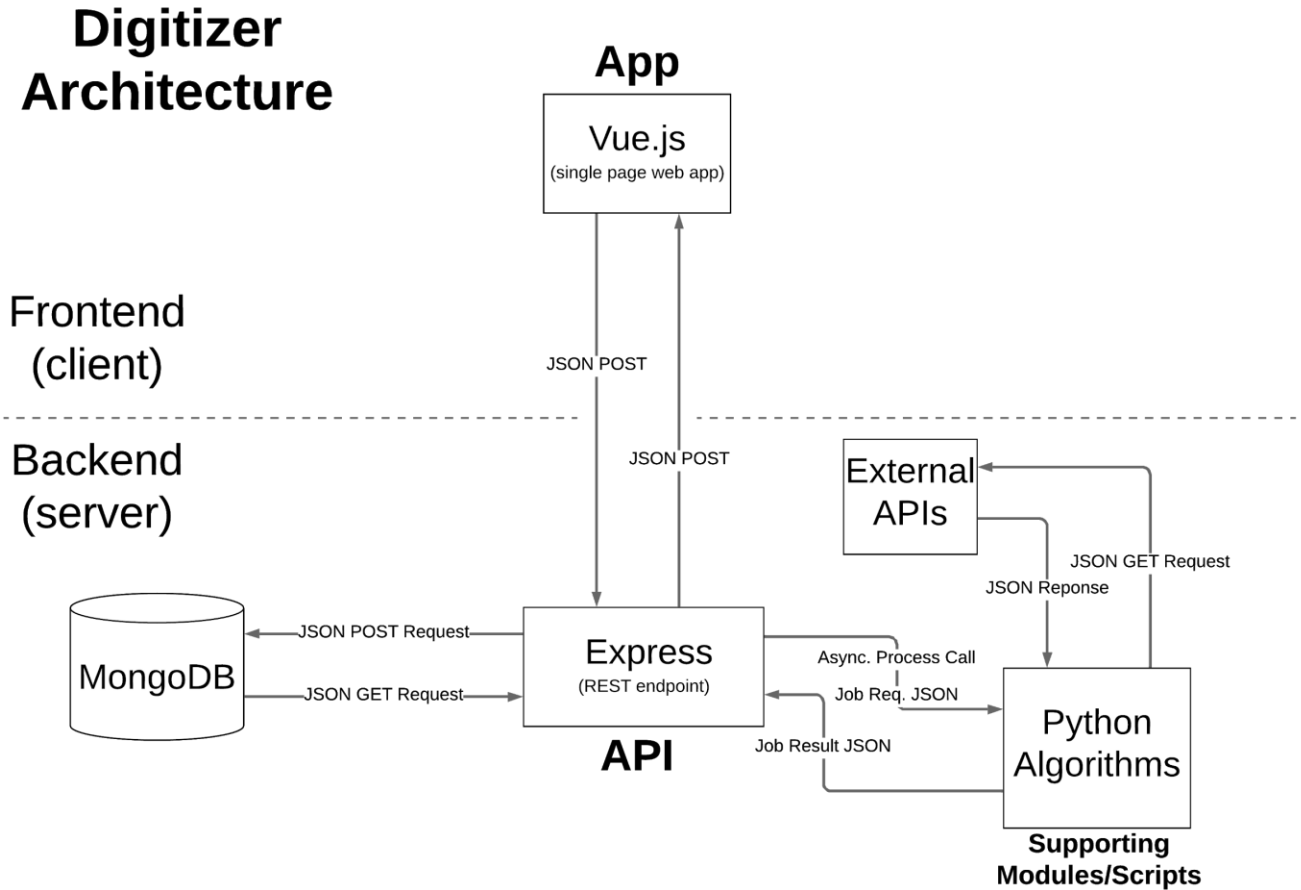


Figure 1: Digitization architecture diagram

The App:

The front-end of our application consists of a single-page web-application implemented in Vue.js, an open-source javascript framework which is extraordinarily fast. Vue.js uses a state store to dynamically manage state across multiple components, allowing applications to update component states without refreshing the webpage, or requiring a bunch of mucking around with the DOM.

The Server/API:

The app will route all its calls to a Node.js/Express server API that manages all the routing between our back end image processing algorithms and the database. Node.js uses an event-driven, non-blocking model to maximize efficiency, and the Express framework has become an industry standard server architecture for scalable web-applications. The API will be passing the image and segmentation data back and forth from the Python scripts to the browser as the app displays the cut-lines for adjustment, the final segmentation, and the results of the OCR. The API will also be responsible for managing storage and retrieval calls from the database.

Database:

All documents, segmented problems, and related data for the app is hosted in a MongoDB database. MongoDB is built on a NoSQL database architecture, meaning that it does not use the relational table model characterizing all SQL databases. NoSQL databases are particularly useful for handling ‘big-data’ and for use in real-time web applications. Mongo has become increasingly prevalent because its unit of storage is a JSON blob, and the database supports indexing as well as field, range, and regular-expression based queries.

Python Algorithms and External API’s:

This component of the diagram represents the scripts and API’s that will be used to perform the initial cut-line proposal, the problem-level segmentation, sub-segmentation, and optical character recognition. While there are many computer vision API’s that provide excellent OCR functionality (MathPix for equations, Google OCR for text), the majority of the segmentation functionality had to be built by tailoring existing algorithms to suit our needs (such as the X-Y Cut algorithm) and combining them with the available functionality provided by open-source computer vision libraries.

Technical design:

Segmentation Research

I. Mean Edge Clustering

Overview

The Mean Edge Clustering Algorithm is a novel approach to problem-level segmentation. The general approach was to utilize the spatial proximity of the elements in a document image to identify the separate worksheet problems.

Steps

A. Edge Detection

First, we used a Canny edge detection algorithm, implemented in Python's OpenCV library, to detect every element's contour in the document image. Each contour can be described as sequence of X/Y Cartesian coordinates.

B. Spatial Averaging

Next, we averaged the X/Y coordinates of each element's contour. This effectively reduced each element contour to an element point in X/Y space.

C. Agglomerative Hierarchical Clustering

Lastly, we used Agglomerative Hierarchical Clustering to spatially group the element points. We chose Agglomerative over other clustering algorithms, such as K-means, for its ability to cluster based on a distance threshold as opposed to a predetermined number of resulting clusters.

Results:

The Mean Edge Clustering algorithm did a reasonable job of segmenting the individual problems. However, some clusters did spill over into other worksheet problems (see the purple data points in figure 2 below.) This finding suggests that, without further refinement, the described approach was suboptimal.

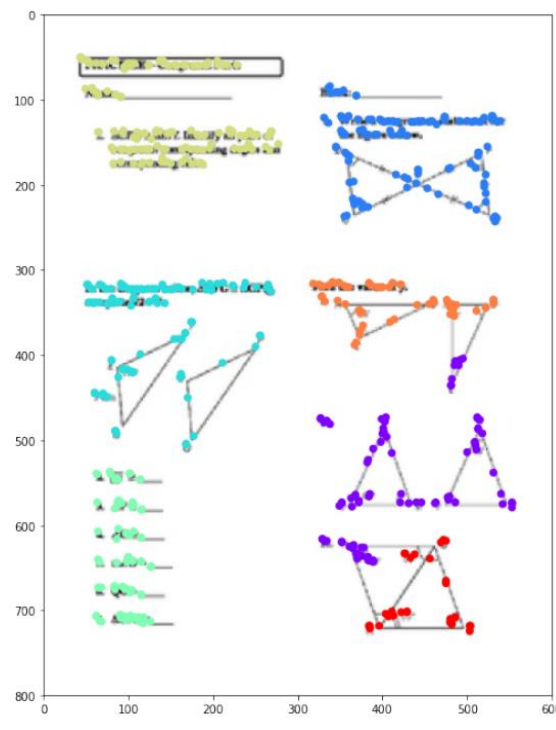


Figure 2: Each color denotes a different segment from the clustering algorithm

II. Gradient Watershed

Overview

The Gradient Watershed algorithm uses elevation descent and a notion of water flow to segment an image into its individual elements.

Steps

A. Elevation Profile

First, we converted the document image to grayscale. At this point, we could think about the image as an elevation map of the document, where the lightest pixels were troughs and the darkest pixels were peaks.

B. Watershed

Then we used a watershed algorithm, implemented in Python's Sci-Kit Image library to model rain falling evenly over the document elevation map. As the water pools in the low lying regions, individual document segments, which are the islands in the water shed analogy, will begin to form.

Results

The Gradient Watershed algorithm offered an improvement over Mean Edge Clustering, but was too granular to perform adequate problem-level segmentation. Additional grouping of the segments could potentially yield more desirable segmentation results. However, we decided that the increased complexity of performing additional grouping operations was enough to justify searching for an alternative solution.

PRACTICE – Congruent Parts

Name: _____ Block: _____

1. $\triangle CPN \cong \triangle HY$. Identify all pairs of congruent corresponding angles and corresponding sides.

2. Write a congruence statement for the figure below.

In the diagram below, $\triangle EFG \cong \triangle OPQ$, complete #3 – 7.

3. $\angle P \cong$ _____

4. $\angle F \cong$ _____

5. $\angle Q \cong$ _____

6. $m\angle O =$ _____

7. $\triangle OPQ \cong$ _____

8. Find the value of y .

9.

10.

Figure 4: Each red box depicts a different object found by Felsenszwalb

Segmentation Implementation:

I. X-Y Cut Algorithm:

One of the most interesting pieces of our project is our algorithm for “task-level” segmentation, which essentially means the segmentation of separate problems of titles. This method (shown in Figure 5) looks at the matrix representation of an imported image for rows or columns of white space and groups consecutive white rows or columns into blocks.

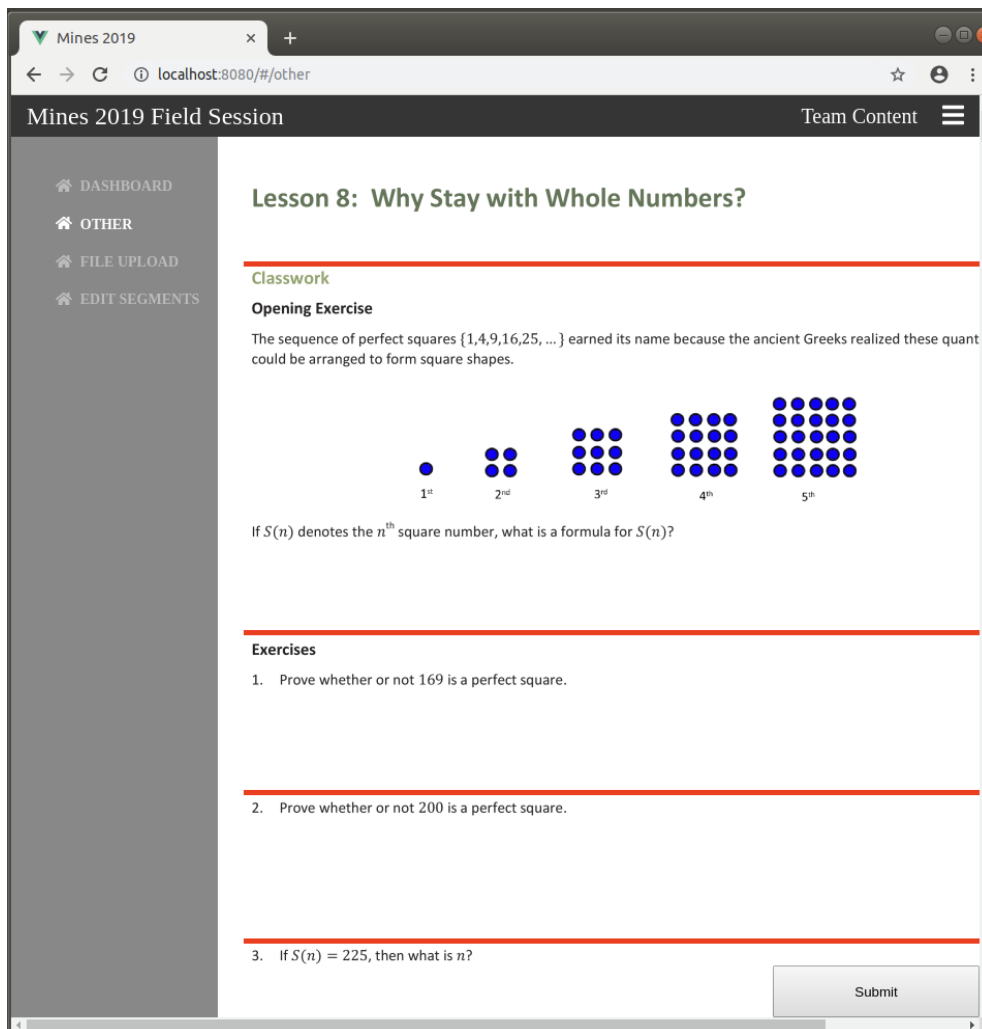


Figure 5: User interface displaying results of X-Y cut algorithm

The mean block size is then calculated for comparison purposes and the algorithm takes any blocks that are within a tolerance of the mean block size. Further filtering is then implemented to confirm that enough segments have been chosen in order to allow for many different document formats. Looking again at Figure 5, note that there are no vertical cuts. This is because there is no columns of entirely white space that meets the threshold requirement to be included as a segmentation option. The vue app, shown in figure 5, takes a JSON blob from the python that contains the locations of each cut-line. We retrofitted the draggable component library with positions that we then set using the values in the JSON. The final locations of the cut lines, adjusted by the user, are sent back to the python for the final cut.

Ethics and Quality Assurance:

A proper discussion of “Quality Assurance” must begin with a definition of quality. In a professional context, we generally define ‘quality’ as a measure of how closely the delivered

product matches the stakeholder needs and desires as described at the outset of the project and in the periodic review sessions conducted throughout the development process. Additionally, there exists a more abstract “ethical” standard that must be upheld, superseding the economic and practical directives of the client, the purpose of which being to protect the conscience, reputation, and integrity of the individual engineer, the profession, and the institutions who have endorsed each contributing entity. To this effect the ACM and IEEE have published standards detailing the proper conduct and priorities of their members, providing guidelines to assist engineers and developers in decision-making processes that shape our experience of technology, life, and ultimately our future. This document describes our ethical considerations in regards to this project, as well as our strategy to ensure that we deliver a product that upholds the standard of quality set by ourselves as individuals, our client Woot Math, and the Colorado School of Mines.

Our quality assurance strategy began with the identification of which particular standards set forth by the ACM are most relevant to our project, and consists of the techniques and tools we will utilize to ensure these standards are met. As contractors, we are obligated to protect the intellectual property of our client, Woot Math. As a team, it behooves us to be open to and accepting of constructive criticism as well as providing appropriate professional review of our work. It is our responsibility, as with all engineers, to work exclusively within the realm of our capabilities and expertise, communicating openly when we are called to expand our knowledge base in order to complete a directive. We hope and desire to create opportunities for growth, teaching and learning in turn, elevating our understanding and that of our team. To meet these goals we address four areas: communication, code, testing, and potential pitfalls.

Communication: Failing to plan, is planning to fail. Understanding client needs requires clear and open discourse. Managing expectations and ensuring equivalent product vision is critical to a successful venture. Exchanging comments and constructive criticism is a crucial aspect of development, beyond the distillation of stakeholder needs. Our team followed the Agile philosophy, holding regular scrum rituals and sprint retrospective reviews to address issues that arose each week as well as to cultivate an environment of accountability. By making time for regular debriefing we could stay on the same page and create space to resolve any challenges and frustrations that arise. All parties must have a clear understanding of the goal, necessitating regular discussion of the current state of the project as well as intermediate goals, and the final definition of done. Our team held regular meetings with one another, our advisor, and our client to keep everyone on the same page.

Code: Obviously, this is the one of the most important sections of quality assurance. We are software engineers. Granted an ethical project, quality code becomes our most highly rated priority. In order to deliver the highest quality code possible we adhered to the style guidelines described by our client, following the principles of object oriented and container-based design, using a linter to catch errors, maintaining open lines of communication to resolve any discrepancies or misunderstandings we may have, and writing clear and concise documentation. Pair programming is a widely renowned technique known to significantly reduce the number of bugs and errors introduced to a code base. We used this technique on both the front and the

back end of our application to reduce the time needed to debug and refine our code.

Additionally, we held code reviews in order to identify problem areas, refactoring as needed.

Testing: Nothing can be deployed at scale without rigorous testing. We used unit tests to check our environment for inconsistencies. Routes must be tested from the app to the REST API, from the API to the database and from the REST API to the Python scripts that will handle the image processing. The Python algorithms that we used to segment and digitize the content must be tested rigorously, as well as all the data meant to be persisted in the database. The functionality of every feature introduced by the development team must be verified. Test driven development is a fantastic technique that can help to accomplish this task. Unfortunately, our clients do not use that particular strategy, however we have written several tests to validate our code to the extent that we can.

Pitfalls: Halfway through a project is quite late to be laying a quality assurance plan. Quality assurance should be discussed at the beginning and at every stage of the project, otherwise the team introduces risks of low prioritization of the plan, overlooking the QA process, and the lack of general ownership of quality. The biggest pitfall of quality assurance is the absence of a cohesive plan all-together. It's important to discuss potential problems before they come up, and for each member of the team to feel comfortable and familiar using the resources available for assistance should difficulties arise. Managing expectations is key, and clear communication so that if the final product falls short of stakeholder desires, at least it will not come as a surprise.

In our exchanges with the client we have identified their need for a basic prototype which they can build upon. Woot Math has communicated that their highest priority is a well documented code base that will be easy for them to build upon and mold into their final product, however the details are somewhat open-ended. Our team constructed an architectural schema and identified the 0th order features we would like to deliver, and both have been approved by the client. This document has outlined our plan to achieve these ends, and we remain optimistic that our standard of work will be upheld. Ethics and quality begin and end with the individual. Taking ownership of these principles fosters an environment of integrity.

“Work is love made visible.

And if you cannot work with love but only with distaste, it is better that you should leave your work and sit by the gate of the temple and take alms of those who work with joy.

For if you bake bread with indifference, you bake a bitter bread that feeds but half man's hunger.

And if you grudge the crushing of the grapes, your grudge distills a poison in the wine.

And if you sing though as angels, and love not the singing, you muffle man's ears to the voices of the day and the voices of the night.”

- Kahlil Gibran, *The Prophet*

Results:

To aid teachers in their transition to the digital classroom, Woot Math envisions a single-page web application that allows teachers to upload and manage mathematical content (worksheets, exams, etc.) for which they requested a prototype. Our initial research offered several reliable options for optical character recognition including MathPIX, OpenCV, and Google's OCR API, however we did not uncover a suitable computer vision library for segmenting math documents. To that effect, our Python team began developing a custom module for recursively segmenting math worksheets using open source libraries, most notably Scikit Learn. This research was fruitful on the first order pass, however sub-task segmentation proved difficult to tune. When segmenting below the 'problem' level the algorithm broke sentences apart in several instances which resulted in errors due to lack of context during the OCR digitization process, prompting our client to suggest implementing an 'X-Y cut' algorithm and a closer look at Tesseract.

The web component of the app was developed in a dockerized, full-stack environment containing a database, a restful API, and a hot-reloading webpage. Our web team was slow to build momentum due to unfamiliarity with a few of the key technologies involved, and while a simple prototype UI is somewhat straightforward to implement, the plumbing involved to stitch the back and front-ends together proved more intricate and painstaking than anticipated. The dockerized environment provided by the client was under development throughout the duration of the project, the final update being a container facilitating calls to link the API with the Python image processing scripts being pushed in the final week of the project. Unfortunately, we were not able to familiarize ourselves with the updates in time to incorporate them. On the bright side, we learned an incredible amount about full-stack application developing in a "real world" scenario. The hot-reloading ability of the web-app, the VSCode tsc watch function, Postman, and the MongoDB GUI tool Robo-3T were all extremely useful in the development process, aiding to illuminate the inner workings of the environment. Being able to poke around under the hood of a web-app from the UI to the DB is a richly informative experience, particularly with the ability to massage the codebase and see instant results. Node.js and MongoDB are quite prevalent in the modern web-dev ecosystem as well, so the opportunity to branch beyond SQL and PHP is quite refreshing (no pun intended, obviously).

By adopting an exploratory and research-heavy approach to this project, we were initially uncertain of how to incorporate our constantly mutating computer vision research results into a cohesive vision for the final web application. Outlining a list of clear features became quite challenging because we weren't sure exactly what the 'sprint' deliverables might look like. Our team struggled to fit our development process to the Agile model early in the project, which cascaded into later weeks. Though we managed to complete several of the pieces involved in what will become a fully functioning application, most notably a suite of computer vision scripts for segmenting and digitizing worksheets, our failure to identify a sequence of attainable and progressive goals each week likely contributed to our inability to include many of the features we had hoped to deliver at the outset of the project. This circumstance certainly lends weight to

the argument for the effectiveness of the Agile model, and acts as an efficacious lesson as to the import of clear communication and the elucidation of intermediate goals to guide the creative process.

Citations

1. <https://www.wootmath.com/>